

5-2008

# An Adaptive Algorithm for Prioritization in 802.11e Wireless Networks

Will Spearman

Clemson University, [willspearman@alumni.clemson.edu](mailto:willspearman@alumni.clemson.edu)

Follow this and additional works at: [https://tigerprints.clemson.edu/all\\_theses](https://tigerprints.clemson.edu/all_theses)



Part of the [Computer Sciences Commons](#)

---

## Recommended Citation

Spearman, Will, "An Adaptive Algorithm for Prioritization in 802.11e Wireless Networks" (2008). *All Theses*. 341.  
[https://tigerprints.clemson.edu/all\\_theses/341](https://tigerprints.clemson.edu/all_theses/341)

This Thesis is brought to you for free and open access by the Theses at TigerPrints. It has been accepted for inclusion in All Theses by an authorized administrator of TigerPrints. For more information, please contact [kokeefe@clemson.edu](mailto:kokeefe@clemson.edu).

AN ADAPTIVE ALGORITHM FOR PRIORITIZATION  
IN 802.11E WIRELESS NETWORKS

---

A Thesis  
Presented to  
the Graduate School of  
Clemson University

---

In Partial Fulfillment of  
the Requirements for the Degree  
Master of Science  
Computer Science

---

by  
William James Spearman  
May 2008

---

Accepted by:  
Dr. James Martin, Committee Chair  
Dr. J. Michael Westall  
Dr. A. Wayne Madison

## ABSTRACT

A common form of local wireless communication is defined by the IEEE as 802.11. Unfortunately, 802.11 has limitations regarding high priority traffic such as voice and data which are sensitive to jitter, delay, and loss. The IEEE 802.11e standard provides enhancements that allow traffic with specific needs to be differentiated from normal traffic. While these enhancements have been shown to effectively improve latency and throughput for high priority traffic, they do not offer precise and consistent control of performance levels for all priorities. In this work, a method to dynamically optimize 802.11e contention parameters is presented that provides more granular control over the network's quality of service for the various data types. A distributed adaptive algorithm that extends 802.11e's Enhanced Distributed Channel Access is presented and is shown to provide more granular and consistent performance than that provided by the static algorithm used in standard 802.11e.

## DEDICATION

This paper is dedicated to Julie. I promise I will turn off the computer now and get some sunshine.

## ACKNOWLEDGEMENTS

The research in this paper was carried out at Clemson University. Funding for research was provided in part under the South Carolina Space Grant Consortium Research and Education Awards Program (REAP). Portions of the research were carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

All software used to produce the research presented in this work belongs to the open source community. I would like to extend a special thanks to those open source projects whose time and energy made this research possible: OpenOffice.org, ns2, Ubuntu, perl, gnuplot, and all the other supporting gnu tools.

## TABLE OF CONTENTS

	Page
TITLE PAGE.....	i
ABSTRACT.....	ii
DEDICATION.....	iii
ACKNOWLEDGEMENTS.....	iv
CHAPTERS	
INTRODUCTION.....	1
802.11 WIRELESS TECHNOLOGY.....	4
THE 802.11 MAC LAYER .....	4
COLLISION MANAGEMENT IN 802.11 .....	5
DCF .....	7
PCF.....	9
QOS IN 802.11E .....	11
EDCA.....	11
HCCA.....	14
OTHER 802.11E ENHANCEMENTS.....	16
CURRENT STATUS OF 802.11E.....	17
PREVIOUS WORK.....	19
PROBLEM STATEMENT .....	24
THE ADAPTIVE ALGORITHM .....	25
METHODOLOGY .....	30
NETWORK TOPOLOGY .....	30
DATA STREAM PROFILES .....	31
METRICS AND DATA COLLECTION .....	33
SIMULATION SCENARIOS .....	34
RESULTS AND DISCUSSION.....	37
PARAMETER DIFFERENTIATION.....	37
BURST SIMULATIONS.....	44
MIXED TRAFFIC SIMULATION .....	51

CONCLUSIONS .....	59
APPENDIX .....	60
REFERENCES .....	61

## LIST OF TABLES

Table	Page
TABLE 1 - 802.11G IFS PARAMETERS .....	6
TABLE 2 - PRIORITY CHANGES .....	29
TABLE 3 – TRAFFIC CLASS PARAMETERS .....	31
TABLE 4 - TRAFFIC TYPES AND ATTRIBUTES .....	33
TABLE 5 – PARAMETER DIFFERENTIATION .....	37
TABLE 6 - BURST SCENARIO DATA .....	44
TABLE 7 - NON-BURST SCENARIO DATA .....	44
TABLE 8 - DATA SUMMARY, AA PRIORITIZATION, MIXED .....	51
TABLE 9 - DATA SUMMARY, STATIC PRIORITIZATION, MIXED .....	51
TABLE 10 - DATA SUMMARY, NO PRIORITIZATION, MIXED .....	51



## LIST OF FIGURES

Figure	Page
FIGURE 1 - 802.11 TRANSMISSION INTERVALS [1] .....	7
FIGURE 2 - CONTENTION FREE INTERVAL [1].....	10
FIGURE 3 - EDCA QUEUE STRUCTURE AT EACH STATION.....	14
FIGURE 5 - SIMULATION NETWORK TOPOLOGY .....	30
FIGURE 6 – THROUGHPUT, PRIORITIZATION WITH AIFS .....	39
FIGURE 7 – DELAY, PRIORITIZATION WITH AIFS .....	40
FIGURE 8 – JITTER, PRIORITIZATION WITH AIFS .....	40
FIGURE 9 – THROUGHPUT, PRIORITIZATION WITH $CW_{MIN}$ .....	41
FIGURE 10 – DELAY, PRIORITIZATION WITH $CW_{MIN}$ .....	41
FIGURE 11 – JITTER, PRIORITIZATION WITH $CW_{MIN}$ .....	42
FIGURE 12 – THROUGHPUT, PRIORITIZATION WITH $CW_{MAX}$ .....	42
FIGURE 13 – DELAY, PRIORITIZATION WITH $CW_{MAX}$ .....	43
FIGURE 14 – JITTER, PRIORITIZATION WITH $CW_{MAX}$ .....	43
FIGURE 15 - THROUGHPUT, STATIC PRIORITIZATION WITH BURST .....	46
FIGURE 16 - DELAY, STATIC PRIORITIZATION WITH BURST .....	47
FIGURE 17 - JITTER, STATIC PRIORITIZATION WITH BURST .....	47
FIGURE 18 - THROUGHPUT, AA PRIORITIZATION WITH BURST .....	48
FIGURE 19 - DELAY, AA PRIORITIZATION WITH BURST .....	48
FIGURE 20 - JITTER, AA PRIORITIZATION WITH BURST .....	49
FIGURE 21 - THROUGHPUT, AA PRIORITIZATION WITH BURST, SINGLE NODE.....	49
FIGURE 22 - DELAY, AA PRIORITIZATION WITH BURST, SINGLE NODE .....	50
FIGURE 23 - JITTER, AA PRIORITIZATION WITH BURST, SINGLE NODE .....	50
FIGURE 24 - THROUGHPUT, STATIC PRIORITIZATION, MIXED.....	52
FIGURE 25 - DELAY, STATIC PRIORITIZATION, MIXED .....	53
FIGURE 26 - JITTER, STATIC PRIORITIZATION, MIXED .....	53
FIGURE 27 - THROUGHPUT, STATIC PRIORITIZATION, MIXED, NO VIDEO.....	54
FIGURE 28 - DELAY, STATIC PRIORITIZATION, MIXED, NO VIDEO_HD .....	54
FIGURE 29 - ROUND TRIP TIME, STATIC PRIORITIZATION, MIXED .....	55
FIGURE 30 - THROUGHPUT, AA PRIORITIZATION, MIXED.....	55
FIGURE 31 - DELAY, AA PRIORITIZATION, MIXED .....	56
FIGURE 32 - JITTER, AA PRIORITIZATION, MIXED .....	56
FIGURE 33 - ROUND TRIP TIME, AA PRIORITIZATION, MIXED .....	57
FIGURE 34 - THROUGHPUT, AA PRIORITIZATION, MIXED, NO VIDEO.....	57
FIGURE 35 - DELAY, AA PRIORITIZATION, MIXED, NO VIDEO_HD .....	58

## INTRODUCTION

In 1997 the Local Area Network/Metropolitan Area Network Committee of the IEEE released the 802.11 standard which defined a wireless standard for data networks in the 2.4 GHz radio frequency range. This original specification set defined communications at a maximum data rate of 2 Mbps, but soon was superseded by two amendments called 802.11a and 802.11b, which defined a maximum data rate of 54 Mbps and 11 Mbps, respectively. These new standards were policed by industry manufacturers united under the name Wi-Fi Alliance, which resulted in greater interoperability among available products and increased consumer adoption. Although 802.11a had higher throughput than 802.11b, 802.11b was the first to market, and gained the greatest market share. The two standards were not compatible due to differences in their physical layer implementations, and hardware that supported both was significantly more expensive than simply supporting 802.11b. As a result 802.11b, and its enhanced version 802.11g, have become some of the most widely used communication standards in the world. As a relatively simple specification using unlicensed radio frequencies, 802.11 gained traction and has become central to the growth of small networks.

Currently, 802.11 devices can be found as standard equipment in most notebook computers for home networking, are used pervasively for corporate internet access, and are even being used by municipalities to provide last-mile internet service to citizens in mesh configurations. However, due to the distance restrictions of using the 2.4GHz band, 802.11 is relegated to small cells of communication such as home computing networks unless mesh deployments are used. Although future specifications such as 802.11n provide greater range, these ranges will most likely not allow 802.11 to function as more than a local communication solution connected to an internet

hub. This small range and relatively low bandwidth compared to wired networks, means that most deployments are likely to have a limited number of stations competing for resources.

In an 802.11 wireless network, stations must share the ability to use the transmission medium, which due to signal loss and corruptions, can be quite difficult. In order to know the current state of other stations that may or may not be transmitting, a station must rely on an error-prone transmission process that is complicated by hidden stations and signal loss. This shared medium access can result in performance problems such as low throughput and high delays in networks with even a small number of active stations. As the number of nodes and amount of traffic increases on a wireless LAN, these issues become more serious in the absence of a central manager. Central management, however, can be complicated and can require additional hardware. While quality of service (QoS) issues have been studied thoroughly in wired LANs, modern high-speed wired networks do not always require complicated QoS due to their relatively high bandwidths, error-free nature, and ease of bandwidth replication due to switching.

Applications that have specific bandwidth or latency requirements do not function well in a congested 802.11 network. VoIP will experience reduced quality during a call when operating over a network experiencing high latencies or low throughput for the connection [8]. Likewise, applications using streaming audio and/or video will experience a reduction in quality when subjected to these conditions. These problems can be more complex than just low throughput and high delay. Large variations in delay can also cause problems with protocols that attempt to adapt to networks with high delay, and in turn cause a considerable degradation in service quality. As wireless technology is adapted to environments containing traffic with different QoS requirements, it is important to find solutions for these issues. A flexible, self-organizing, yet simple network that is able to differentiate between priority services such as the one presented in

this work will enhance the operability of network deployments that are complex and involve large numbers of mobile nodes as well as small networks with only a few critical services.

The IEEE 802.11e standard provides enhancements that allow traffic with specific needs to be differentiated from best-effort traffic. While these enhancements have been shown to effectively improve latency and throughput for high priority traffic, they do not offer precise control of performance levels as network conditions change. For example, in a network with a single high priority node and a single low priority node where both nodes have sufficient resources to achieve all required service levels without differentiation, the low priority node is unnecessarily penalized. More importantly, it will be shown that networks are dynamic, resulting in optimal configurations that are also dynamic and require more logic than static prioritization. While starvation and other downsides of static prioritization could be viewed as acceptable considering the low priority traffic by definition is not important, optimal network conditions exist that allow both high and low priority traffic to find a balance where both experience acceptable service while preserving their priority.

In this work, a method is presented to dynamically optimize 802.11e contention parameters to provide more granular control over the network's quality of service (QoS). A distributed adaptive algorithm that extends 802.11e's EDCA is presented. We show that the enhancement provides more granular and consistent performance than that provided by the static algorithm used in standard 802.11e.

## 802.11 WIRELESS TECHNOLOGY

The 802.11 set of standards covers telecommunications and information exchange between systems in local and metropolitan area networks [1]. The 802.11 standard provides best-effort packet services for the Medium Access Control (MAC) layer of wireless networks. This MAC layer provides wireless stations with fair access to the medium in a best-effort manner. Control of the medium can be centrally managed, or can be left to the individual stations. In the following subsections the base 802.11 MAC protocol is reviewed, followed by the improvements added by 802.11e.

### The 802.11 MAC Layer

The 802.11 MAC layer is built around two coordination functions that control medium access by either distributed coordination or centralized coordination. These two functions compose a super-frame that governs medium access first in a contention-free manner, and then in a contention-based manner. In the contention-free portion known as the Point Coordination Function (PCF), control is centralized to a single point that is usually the Access Point (AP), and in the contention based portion known as the Distributed Coordination Function (DCF), the access control mechanisms are located at the station. The PCF half of the super-frame is optional and is not used in the absence of a central controller. Both functions have advantages and disadvantages and are each suited to particular situations.

### Collision Management in 802.11

In 802.3, commonly known as Ethernet, the primary method for medium access is Carrier Sense Multiple Access with Collision Detection (CSMA/CD) in which collisions are detected on the channel, and are handled by back-off counters that reduce future collisions by randomly increasing window sizes. Detection and recovery are efficient and feasible in wired networks due to high bandwidth, low packet times, the ability to transmit and receive simultaneously, and abundant switching. This reliable nature of wired networks significantly reduces the number and impact of collisions, which directly increases the efficiency of the network by decreasing the number of management frames being transmitted. In the wireless realm, however, interference can cause substantial noise resulting in frequently corrupted packets. Also problematic is the fact that channel sensing is not possible since most radios can not simultaneously send and receive [1]. Even without these issues, the fact remains that each wireless station must contend for the right to transmit into a limited number of shared frequencies. For these reasons collision detection is not possible for 802.11 wireless networks, motivating the need for CSMA with Collision Avoidance (CSMA/CA).

CSMA/CA works on the principle of listening before transmitting. By using wait times efficiently collisions can be minimized and all stations can be allowed to gain access to the medium in a relatively fair manner using DCF or PCF, or both. This algorithm relies on inter-frame spacing to coordinate the communication of the stations. Inter-frame spaces (IFS), or, the time intervals between frame transmissions, are dependent on physical specifications such as propagation delays and radio receiver switching. The four primary inter-frame spaces are SIFS, PIFS, DIFS, and EIFS. Examples of these intervals are shown in with the 802.11b compatible mode shown on the left and pure 802.11g mode settings on the right.

IFS	Mixed Mode	Pure Mode
Slot Time	20 $\mu$ sec	9 $\mu$ sec
SIFS	10 $\mu$ sec	10 $\mu$ sec
PIFS	30 $\mu$ sec	19 $\mu$ sec
DIFS	50 $\mu$ sec	28 $\mu$ sec

Table 1 - 802.11g IFS Parameters

The SIFS, or Shortest Inter-Frame Space, is the time between the last transmission and high priority transmissions such as Clear-To-Send (CTS) frames, positive acknowledgments (ACKs), continuation frames in a burst transmission, and polling responses. Positive ACK frames are given priority so that a station that has just completed reception of a frame can give the sender immediate feedback. RTS and CTS frames coordinate communication between pairs of stations so that other stations know to expect the medium to be free for the duration of the exchange. For this reason, these control frames have precedence over normal data transmissions. If a station completes a frame transmission and has enough time left in the allotted transmission window to send an additional frame, it is allowed to send after a SIFS.

The PIFS, or PCF Inter-Frame Space, is the minimum time a point coordinator must wait before transmitting a frame that begins the polling sequence. This interval allows the point coordinator to begin transmitting before other stations. The DIFS, or DCF Inter-Frame Space, is the minimum time any station other than the coordinator must wait to be able to transmit. When the medium has been sensed as free, and after a DIFS, a station may begin decrementing its back-off counter. The PIFS is shorter than the DIFS so that the central coordinator can take control of the network at any time. The EIFS, or Extended Inter-Frame Space, is the minimum wait time for a station that receives corrupted frames or other errors. The EIFS is often variable depending on the type and number of errors. The EIFS time is designed to give another station

time to ACK the frame that was received and interpreted as corrupt, before the station has the chance to transmit again. If a valid frame is received before the EIFS has expired, the EIFS timer is cancelled since normal synchronization with the network has occurred.

The frame spacing intervals allow DCF and PCF to interact seamlessly and with as few collisions as possible by always assuming the following relationship:  $SIFS < PIFS < DIFS < EIFS$ . In Figure 1 taken from [1], it is shown that frames governed by smaller time delay intervals will have a distinct advantage by being able to transmit before those with longer intervals. In the following sections, it will be shown how these inter-frame spaces are the foundation for providing control mechanisms with priority over stations, as well as providing station priority in the absence of centralized control.

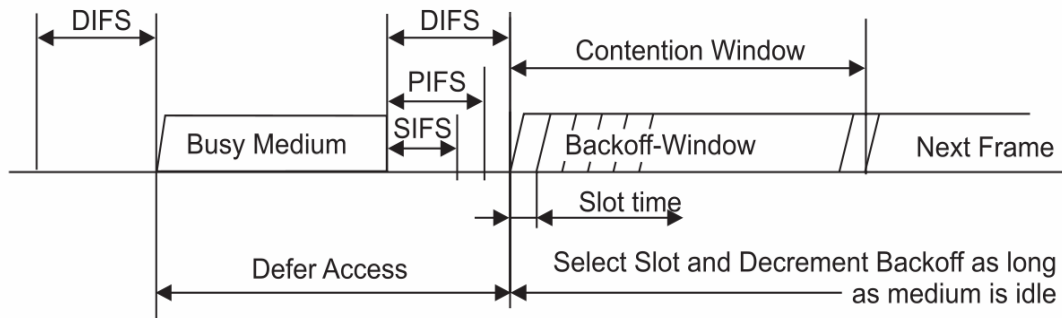


Figure 1 - 802.11 Transmission Intervals [1]

### DCF

DCF allows stations to transmit without a central coordinator by using a back-off timer. When a station wishes to transmit, and has sensed that the medium is free it waits for a DIFS and then checks its back-off counter. If the counter is zero it transmits, or if the counter is non-zero it



decrements the counter for each slot time that passes. The back-off counter value is randomly picked from a window between zero and the Contention Window (CW), which begins equal to  $CW_{min}$  and is doubled after each consecutive collision to a maximum value of  $CW_{max}$ . Each time the transmission fails, a retry counter is incremented. If the retry counter reaches the retry limit, the packet is dropped. If the transmission succeeds, the CW is reset to  $CW_{min}$ , and a new back-off counter is chosen. If during the DIFS, the medium becomes busy, it defers transmission until the medium become free again. As collisions occur and the back-off counter exponentially increases, it has the effect of reducing contention in the network by reducing the frequency of attempts to transmit, and at the same time increases the probability that a station that is ready to transmit will not cause another collision.

DCF also includes an optional RTS/CTS mechanism to eliminate the hidden station problem. The hidden station problem occurs when two stations can sense transmissions of the AP, but not of each other. Due to their inability to receive each other's signals, the two stations can claim the medium simultaneously, and will cause a collision at a central destination. To prevent this, before sending a frame, a station transmits a RTS packet and then receives a CTS packet from the central station. Both of these frames include information regarding the time it will take to send the frames, which allows other stations to set a timer called the Network Allocation Vector (NAV) since the medium will be busy at least for that length of time. The NAV works like a virtual carrier sense in that if a station has a positive value for its NAV it knows the medium will be busy without physically sensing the medium. After the NAV expires, stations begin normal time interval waiting and back-off counter decrementing. Since CTS frames are allowed to be transmitted after a SIFS, they have priority over normal DCF transmissions.

## PCF

In PCF, medium access is controlled by a Point Coordinator (PC). The PC controls access by querying stations wishing to transmit by polling them during a Contention Free Period (CFP). Together the DCF CP and the PCF CFP form a superframe which repeats for each time period. During the CFP, PCF is used to control access, and then during the CP, DCF is used.

The CFP portion of the superframe begins with a beacon frame that contains management information such as protocol parameters and time synchronization. The time synchronization includes a time value that stations use to set their NAV for the duration of the CFP so that no station attempts DCF transmissions. After the beacon frame has been transmitted, the PC polls stations in a round-robin manner, and upon successful response, allows the station to transmit either an ACK indicating it has nothing to send, or a DATA+ACK frame. Having received no response from a station, the PC moves on and the station is not allowed to transmit until the CP, or during the next CFP. The CFP ends when the time period specified by the beacon frame expires, or a CFP-EndFrame is sent. After the CFP has ended, a normal DCF period proceeds. However, since PIFS is shorter than DIFS, the PC can immediately seize the medium and begin another CFP if desired.

Figure 2 shows a typical exchange during the CFP. Once the medium has been free for a PIFS, the PC sends the beacon frame which includes the NAV value. During the CFP, all responses to the PC polls are transmitted after a SIFS just as normal ACK frames are, but for efficiency ACKs can be combined with data and polls such that an entire CFP sequence can use SIFS. As Figure 2 shows, the only time PIFS is used again is when a station fails to respond to a poll. Since not all stations may be able to be polled or may not have data to send, an unanswered

poll frame that lacks data may not be answered. Compared to a CP using DCF, the CFP shown is a more efficient usage of the medium since collisions are unlikely, and since SIFS is used between most frame transmissions. PCF becomes less efficient when many stations are present and overhead from polling becomes significant with respect to the number of stations that are ready to transmit.

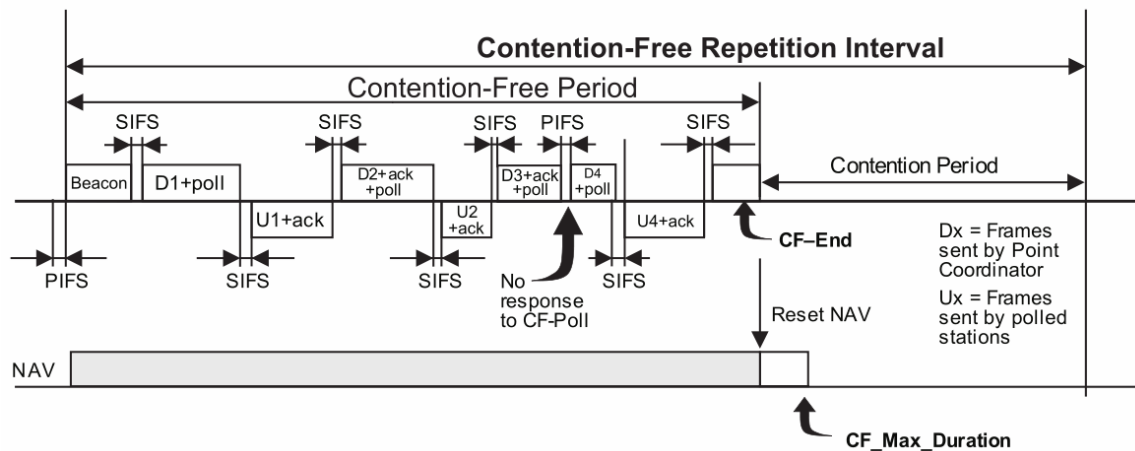


Figure 2 - Contention Free Interval [1]

While PCF was intended to provide a form of QoS to 802.11 networks, it is generally agreed that it fails to provide this service adequately. Although PCF gets priority over DCF since the PIFS is always less than the DIFS, it suffers from the fact that individual network flows cannot be singled out for prioritization since the PC polls in a round-robin fashion. High priority can be given to individual stations by marking some stations as pollable and others are non-pollable, but affecting service on a more granular level is impossible with PCF. Also, polling can result in excessive overhead and large end-to-end delay when the number of stations is large [8].

## QoS IN 802.11E

In an effort to give 802.11 networks true QoS, 802.11e was standardized in 2005. 802.11e introduced enhancements to the existing DCF and PCF, placed them under the heading of the Hybrid Coordination Function (HCF). The HCF is comprised of Enhanced Distributed Channel Access (EDCA), which is an enhanced DCF, and HCF Controlled Channel Access (HCCA), which has many traits in common with PCF. These two access methods work separately or together, just as in 802.11, where DCF is mandatory and PCF is optional. While the fundamentals of the original functions were not changed, augmented information allows HCF to provide QoS to specific flows and/or stations.

### EDCA

In the EDCA, MAC layer parameters are used to provide priority to each traffic class (TC) in a contention access manner similar to DCF. The parameters that can be manipulated are the Arbitration Inter-Frame Space (AIFS), the Transmission Opportunity (TxOp), the  $CW_{min}$ , and the  $CW_{max}$ . These parameters are given default values at each station for each TC, or they can be overridden by a PC using special coordination frames.

As shown in Figure 3, the MAC layer queue structure in the EDCA is composed of multiple DCF queues. In DCF, each station has a single queue that all attached traffic sources traverse in a first-in-first-out manner, where the probability of a source being the next to transmit is dependent on the packet size and packet rate of that source. In the EDCA, a station has for each TC: a DCF queue, prioritization parameters, and a transmission packet. Each transmission packet at each queue head contends for the right to transmit by decrementing a back off counter based on the

parameters for its queue. When more than one queue reaches a back off counter value of zero, the transmission packet from the highest priority TC is selected for transmission and the other queues experience a virtual collision. This virtual collision is treated as if a real collision has occurred; the contention window for each queue will be doubled and a new back off counter will be picked.

The priority of one queue over another is dependent on how long it must wait before being able to transmit, and the length of time a packet must wait to be transmitted is controlled by the back off counter. The AIFS parameter contributes to this time by defining the interval the TC must wait before starting the back-off counter after the medium has been sensed as free. The  $CW_{min}$  and  $CW_{max}$  parameters prioritize by adjusting the back-off counter's minimum and maximum sizes, respectively. Each parameter can be varied within TCs, while normal DCF rules apply between TCs within a station. The  $TxOp_{max}$  defines the length of time a queue may transmit. A longer TxOp value contributes to higher throughput and can reduce overall contention in the network by consolidating packets of bursty traffic sources.

In EDCA, the AIFS is a time interval that is equal to or greater than the DIFS such that higher priority stations can be given low values. When the AIFS expires, normal DCF operation for a queue continues by decrementing the back-off timer. Therefore, TCs with low AIFS values will be more likely to gain access to the medium since their back-off counters will begin decrementing sooner. As a queue experiences collisions and the CW grows, the AIFS value becomes an increasingly small portion of the total time it must wait to transmit. For this reason AIFS is most influential when CW is equal to  $CW_{min}$ .

The  $CW_{min}$  value controls the size at which each TC starts its CW. A lower  $CW_{min}$  allows the queue to seek the medium sooner after the AIFS as long as the previous transmission was a success and the CW has not been increased. For each transmission attempt after the first for a particular frame, the  $CW_{min}$  value is less irrelevant since it increasingly has less influence on the value of the back-off counter. When a collision does occur, each TC multiplies the current CW by the Persistence Factor (PF, usually equal to 2) to calculate the next back-off timer.

The  $CW_{max}$  controls the maximum value to which a flow's CW can grow. A larger value will allow the flow to be less competitive during heavy collision, high load situations. Lower priority flows will have larger CW values and will wait longer when network traffic is causing many collisions. Depending on the  $CW_{min}$  value, a queue with a smaller  $CW_{max}$  value is more likely to hit the retry limit sooner.  $CW_{max}$  only influences performance when the queue has experienced many collisions.

Finally, the  $TxOp_{max}$  controls the length of time for which a station can transmit on behalf of each TC. Larger  $TxOp_{max}$  values allow stations to send more frames during each use of the medium. This parameter influences performance most for queues with high data rates such as video.

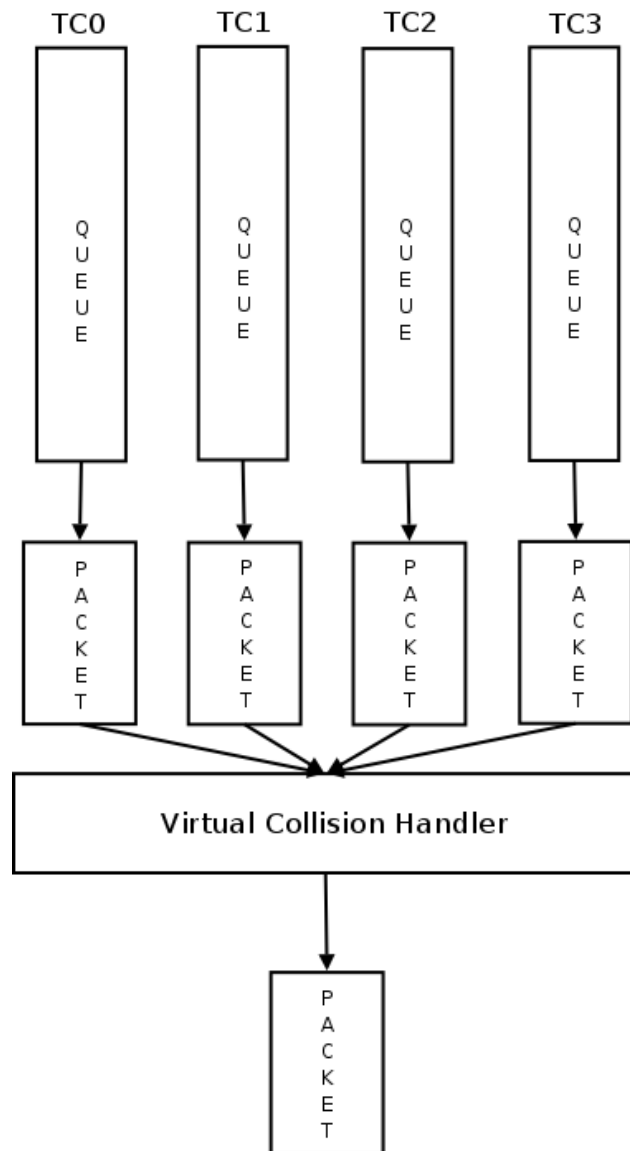


Figure 3 - EDCA Queue Structure at Each Station

### HCCA

In the HCCA portion of the HCF, a Hybrid Coordinator (HC) is used to centrally manage the medium in much the same way as PCF, with the exception that HCF uses parameterized QoS, and may initiate a CFP at any time. Parameterized QoS refers to the use of specific information,

called the Traffic Specification (TSPEC), such as minimum data rate, maximum latency, specific parameter values for each TC, etc. that allows the HC to prioritize accordingly. When acting on behalf of a new data stream, the station sends its requirements to the HC for acceptance. The stations transmit these requirements in the form of TSPEC frames to the HC, which can accept or reject the request based on network conditions.

A distinct advantage of HCCA over PCF is the ability to provide granular priority based on data streams rather than stations. A HC can choose to poll a queue from a station rather than the entire station. In PCF, the polling is based on round-robin rotation, but in HCCA polling is based on the admitted TSPECs from stations. The logic for selection, polling, and priority are left to the implementation, and therein lies a disadvantage of HCCA; the policies to control the admission of new streams, acceptance of requested QoS from data streams, fair polling, and frequency of CFPs is largely up to algorithms implemented on the HC rather than built-in specifications. Setup of an HCCA network is complex.

Similar to the relationship between DCF and PCF, EDCA and HCCA are integrated within 802.11e yet operate independently. Even when HCCA is not used, EDCA uses superframes with CFPs and CPs. If an HC does not exist, or chooses not to poll during a superframe, the CP starts and EDCA rules are followed. The HC's transmission is governed by the PIFS time interval, and since this interval is smaller than the AIFS, the HC can obtain the medium before normal stations do. If the HC is active, it polls stations that have indicated a need to be scheduled, allows them to transmit, then ends the CFP with a CF-End control frame.



### Other 802.11e Enhancements

The 802.11e standard defines the HCF in order to provide specific data streams and stations with high priority over others, but also defines other mechanisms to indirectly aid in this goal, such as: Contention Free Bursts (CFB), Special ACKs, and Direct Link Protocol (DLP).

Under normal operation, a station must contend for channel access after each TxOp. This can result in low throughput for data-rate sensitive applications even in the situation where it is given priority over other stations. More importantly, this reduces network utilization by dividing the data stream into separate TxOps. Using CFB, if there is still time left in the TxOp after a frame is transmitted, the station is allowed to continue after waiting for a SIFS. By using CBF in conjunction with higher  $TxOp_{max}$  values, CFB can reduce the amount of overhead and allow a high priority station that already holds the medium to achieve greater throughput.

The second indirect improvement is the addition of two options to the QoS control field of data frames regarding ACKs. A station has the option to send packets with a flag set such that an ACK for that packet is not generated. In order to increase efficiency the NoAcknowledgement (NOACK) flag can be used for applications where ACKs are not important, or do not signify any action on the part of the sender. Streaming media, for example, can tolerate packet loss but suffers greatly in the event of high latency, which would be the result of multiple retransmissions due to a lost ACK. NOACKs in this situation improve efficiency by eliminating unnecessary acknowledgement for the receiver since no retransmission attempt will be made for real-time data. A second type of ACK called the BlockACK is also defined as aggregated ACKs that accumulate during a CFB. An ImmediateBlockACK can be requested by the sender after a CFB, and if necessary, the receiver can respond with a DelayedBlockACK if the receiver cannot respond before the sender's request

timeout. These ACKs allow CFBs to be used effectively in an 802.11e environment, but do not directly improve prioritization. They do, however help to offset the added contention overhead of 802.11e.

Basic 802.11 operations allow ad-hoc networking directly between stations or infrastructure-based networking where stations cannot communicate directly. However in 802.11e DLP allows networks to use both simultaneously. When a station wishes to use DLP, it sends a DLP request to the AP. The AP forwards the request to the receiver. If the receiver supports DLP it will send a response through the AP back to the sender. The sender will then contact the receiver directly and begin the transmission. This direct communication can be especially useful when two stations are located near each other, but distantly from the AP. The signal between the two stations may be stronger and could result in fewer dropped frames. More importantly, eliminating the extra step of going through the AP can reduce the round trip times by half and lower the load on the AP for forwarding unicast traffic. The DLP feature could be very desirable for supporting local voice loops and potential video streaming between two 802.11 stations.

#### Current Status of 802.11e

The lack of clear market requirements in the consumer space for 802.11e, and additional product complexity has resulted in few vendors shipping a full 802.11e implementation [3]. In an effort to stimulate development towards full 802.11e deployment, the Wi-Fi Alliance (WFA) developed requirements for hardware to be Wi-Fi Multimedia (WMM) compliant and Wi-Fi Scheduled Multimedia (WSM) compliant [3]. These two standards are subsets of 802.11e that allow stations to support consumer applications that would benefit from prioritization. WMM uses EDCA while WSM uses HCF, but neither includes the other enhancements discussed in the

previous sections. WMM uses four categories in which to place traffic: voice, video, best effort, background. Each of the eight 802.11e categories is mapped evenly to a traffic class to allow backwards support for non-WMM stations. Any station sending data that is not assigned a traffic class is considered best effort traffic.

## PREVIOUS WORK

Previous work in the area of 802.11 QoS has shown that 802.11 network parameters can be adapted to provide better overall network service to general clients by maximizing throughput based on current network conditions [6, 7]. It has also been shown that existing 802.11 designs allows parameter tuning such that services like VoIP can be accommodated and given some QoS guarantees [8]. The fact that these approaches do not accommodate multiple priority levels or standardized parameters has lead to the development of 802.11e. Work has been done to dynamically adjust 802.11e parameters to provide better performance and QoS. These developments, and their relevance as the basis of an adaptive algorithm, are discussed in the remainder of this section.

In [16] the effects of various contention window sizes are explored. The authors show that the default value of  $CW_{min}$  in 802.11 leads to under-utilization of the network as network conditions change. They experiment with other values, finding that in networks with small numbers of stations, small  $CW_{min}$  values greatly increased the collision probability, reduced overall throughput, and increased delays. However, as the number of stations increased and the network became fully utilized the penalty was diminished, since the probability of a station being ready to transmit when the medium was free was increased. The authors also point out that when a single station is transmitting, throughput is greatly increased when  $CW_{min}$  values are small since the medium is idle less time.

In [6, 7] the authors evaluate a mechanism which allows dynamic tuning of the timing used in the back-off algorithm in 802.11. They showed that a dynamic algorithm based on the number of

currently active stations, which manipulates the minimum back-off time can allow a wireless network to perform closer to the theoretical capacity of the medium. The algorithm estimates the current performance and adjusts transmission probabilities. Their findings show that static network parameters lead to under-utilization of the medium and show the importance of a dynamic algorithm.

In [10], the importance of dynamic solutions to service prioritization becomes clear in the results presented. It is shown that the design of 802.11e increases the possibility of collisions and increases delay by adding an extra contention layer. In normal 802.11 networks the only collisions that can occur are those between stations as they try to gain access to the medium to transmit the packet at the head of the single send queue. In 802.11e, virtual collisions can occur when each TC queue must contend for access within the station's queue manager, as well as real collisions when the winning queue is allowed to transmit. The results in [10] show that throughput in 802.11e is decreased and latency is increased when compared to 802.11a networks due to this extra contention being introduced. Therefore, dynamic and aggressive tuning of the network parameters is required to achieve benefit from prioritization.

In [18] a method for dynamically tuning 802.11 is shown using admission control and service differentiation. The admission control portion of the method is intended to keep the network stable when a new source is introduced. In order to achieve service differentiation, the authors modify the Access Point to operate on an intelligent round robin approach to TxOp allocation. The optimizations show reduced loss rates, and increased throughput.

In [4], the authors evaluate how a network with 802.11b and 802.11e nodes performs with different EDCA contention parameters, and how the delay and throughput are affected by these

parameters. The 802.11b nodes model background traffic while the 802.11e nodes model high priority traffic. Four different contention parameters are tested: the initial size of the contention window,  $CW_{min}$ , the maximum size of the contention window,  $CW_{max}$ , the Arbitration Inter-Frame Spacing interval (AIFS), Persistence Factor (PF). These parameters can be adjusted to differentiate 802.11e traffic from 802.11b traffic present on the same network.

In [4], the authors study the individual affects of each 802.11e parameter on prioritization. AIFS was shown to be the most effective contention parameter for protecting high priority traffic from background traffic. However, the authors show that using PF and  $CW_{min}$  for differentiation may have the advantage of allowing for better performance of the low priority traffic. The  $CW_{min}$  parameter can be characterized as a compromise between AIFS, which is the most effective for high priority nodes, and  $CW_{max}$ , which is the least adverse towards low priority nodes.

The work presented in [11] is similar to our research. The authors use a two-level approach to fair, yet prioritized service. The first level of protection for high priority services guarantees that changing network conditions do not affect data streams such as VoIP and video that have constant QoS requirements. By using budgeted TxOp values for each queue, new flows are not allowed to have immediate access to their share of bandwidth regardless of their TC or parameter settings. This ensures that established flows are not disrupted by new flows. The second level of protection, called Fast-Backoff with Dynamic Adjustment when Fail or Successful, is the most similar to our work due to its distributed nature. Under this scheme, when a station experiences a transmission failure, its CW is increased by a factor greater than 2 which results in a faster than exponential backoff. In addition to the CW increase, the station's  $CW_{min}$  is increased by a factor. When the station experiences a transmission success,  $CW_{min}$  is decreased by a factor, and the CW is reset to  $CW_{min}$ . This dynamic adjustment results in a dramatic decrease in the number of

collisions, as well as more reliable service for the voice and video data. This method adjusts the EDCA parameters on all successes and all failures with no regard for direct network performance measurements. The approach in [11] differs from the approach presented here in that the Adaptive Algorithm uses direct network performance measurements as opposed to using transmission successes and failures. Also, [11] uses the default 802.11e TCs under normal circumstances, and does not allow fine grained performance control.

In [9], 802.11e parameters can be tuned based on network conditions to allow better performance than a single setting. Although these settings are not changed dynamically in this study, they do show that changing network conditions require changing parameters to use the channel efficiently. These optimizations are evaluated in a test-bed under realistic conditions. The authors investigate two methods of choosing  $CW_{min}$  in 802.11e networks based on proportional fairness and time-based fairness. They conclude that proportional fairness in a network based on weights provides higher throughput than time based fairness. Their work shows that  $CW_{min}$  can be tailored to a network to provide all nodes with fair access to the medium if priority mechanisms are used, and therefore it follows that the same principles can be used to provide nodes with unfair access, or differentiation, using contention window parameters.

Studies to improve specific types of traffic using tuned 802.11e parameters have been explored in [17] and [19]. In [17] fairness and throughput are improved by prioritizing based on traffic type and traffic direction. They observe that when many stations transmit, but all receive from a single access point, the access point becomes a point of congestion since it will get an unfair portion of transmission opportunities compared to its importance. In [19], the EDCA is used to prioritize TCP ACK packets. Since lost ACK packets result in retransmissions, and therefore

more packets on the network, ACKs are given higher priority than data packets. Not only does this approach improve network utilization, it is possible to differentiate based on this method.

In works such as [13], the HCCA is shown to be better at efficiently using the medium than pure EDCA. Contention in HCCA networks is reduced since the central coordinator controls access in an organized manner. Although the stations still use EDCA at the station level queues and during CPs, the central coordinator controls most of the transmission opportunities. Despite this result, EDCA can still be considered to be an important research topic from a reliability perspective. In the HCCA scheme, the AP is a single point of failure. If the AP fails, the network will fall back on Ad-hoc mode with EDCA controlling access.

To the author's knowledge no other work exists that dynamically adjusts AIFS,  $CW_{\min}$ , and  $CW_{\max}$  independently using local performance feedback for decision making. Using the works previously discussed, a new method for prioritization is presented based on this premise, and is explored in the following sections.



## PROBLEM STATEMENT

To summarize:

- It is difficult to know the optimal value for network configuration parameters because they depend on changing network conditions.
- Since network conditions change, static configurations can lead to unsatisfactory performance for some stations and data streams.
- Coordinated access is more efficient, but is more complicated to implement and contains a single point of failure, and in the event of failure distributed access is used.

The algorithm presented in this paper attempts to solve the above listed problems by adjusting the EDCA parameters independently for each TC at the station based on a single value assigned to each TC. This single value determines the relative performance of the TC compared to the other TCs, and is adjusted dynamically based on local performance feedback. The algorithm is relatively simple to implement, and provides priority to all TCs. The following sections explain the methods used to accomplish this goal.

## THE ADAPTIVE ALGORITHM

The Adaptive Algorithm (AA) tunes the AIFS,  $CW_{\min}$ , and  $CW_{\max}$  parameters based on the performance of the network, which is measured locally using a measure referred to as the delay factor (DF). The DF is the ratio of the average access delay to an estimate of the minimal access delay. The minimal delay, or the optimal delay, is defined as the delay a TC experiences as the sole transmitter in the network. The DF is a measure of how well the traffic from one TC is performing relative to its optimal performance and assesses the amount of contention the queue is experiencing due to other competing queues or stations. The DF is a measure of the additional overhead created by multiple queues and stations competing for the medium.

During algorithm operation, a DF is calculated for each measurement interval and depends on the performance of the TC for that timer interval. The DF is calculated using the average MAC access delay for the current time period, and the most recently calculated optimal delay. A fixed value called the DF\_TARGET is the overall performance goal of the TC, while the DF is the measure of the progress toward that goal at a particular point in time. The algorithm works by adjusting the queue parameters based on this feedback and goal pair. Figure 4 shows the pseudo-code that corresponds to the algorithm discussed in the rest of the section.

The main loop on lines 39 – 50 of Figure 4 is executed by each TC queue in the MAC layer of each 802.11e station. This loop is executed for each frame sent, and the two procedures, OPTIMAL and AA, are executed conditionally based on time intervals. Each TC independently keeps track of access delays, parameter values, time intervals, tolerance, and other values used in managing these primary values.

```

01.  PROCEDURE OPTIMAL
02.      aifs = AIFS_OPTIMAL
03.      cwmin = CWMIN_OPTIMAL
04.      cwmax = CWMAX_OPTIMAL
05.      FOR OPTIMAL_SAMPLES
06.          optimal_delay += delay
07.      ENDFOR
08.      optimal_delay /= num_samples
09.      aifs = AIFS_TC2
10.      cwmin = CWMIN_TC2
11.      cwmax = CWMAX_TC2
12.  END PROCEDURE
13.  PROCEDURE AA
14.      df = interval_delay / optimal_delay
15.      df_ratio = df / DF_TARGET
16.      IF df_ratio in range(ACCEPTABLE)
17.          increase_tolerance()
18.          return
19.      ELSE
20.          decrease_tolerance()
21.      ENDIF
22.      IF tolerance > TOLERANCE_TRIGGER
23.          return
24.      ENDIF
25.      IF df_ratio in range(LOW)
26.          IF df_ratio < LOW
27.              aggressive_priority_decrease()
28.          ELSE
29.              relaxed_priority_decrease()
30.          ENDIF
31.      ELSEIF df_ratio in range(HIGH)
32.          IF df_ratio > HIGH
33.              aggressive_priority_increase()
34.          ELSE
35.              relaxed_priority_increase()
36.          ENDIF
37.      ENDIF
38.  END PROCEDURE
39.  LOOP MAIN
40.      interval_delay += current_delay
41.      IF now > aa_timeout
42.          interval_delay /= num_samples
43.          CALL AA
44.          interval_delay = 0
45.          aa_timeout = now + INTERVAL
46.      ELSEIF now > optimal_timeout
47.          CALL OPTIMAL
48.          optimal_timeout = now + INTERVAL
49.      ENDIF
50.  END LOOP

```

Figure 4 - Adaptive Algorithm Pseudo-code

The first primary concern for the algorithm is providing a way for each queue to derive information about the system's performance based on local observation rather than global knowledge. This goal is achieved using the mean access delays to characterize the current performance compared to the DF\_TARGET. Based on this performance relationship, tuning

decisions can be made. During operation, the algorithm uses mean access delay samples from before and after changes to the 802.11e parameters (Figure 4, line 40) in order to measure success or failure of the previous evaluation period's parameter adjustment.

The second important design decision involves finding a reliable method of providing a TC with knowledge of its overall goal for its access time, and more importantly, knowledge of when to give up pursuit of that goal. As a solution to this, a minimal mean access delay is measured when the queue comes online, and periodically during normal operation (Figure 4, Procedure OPTIMAL). During this optimal measurement phase, a queue is given highest priority in order to estimate the minimum queue delay. For this short period, the queue lowers its AIFS,  $CW_{min}$ , and  $CW_{max}$ , and assumes highest priority to get a "best case" measurement, to which it will then compare its mean access delays gathered later. This measurement is not intended to measure any physical or link layer capabilities, and therefore does not include ACK times. Instead, it is intended to estimate the performance possible if the medium were not being shared with other stations and queues. This optimal delay is a measure of the queuing delay overhead resulting from contention with other queues and other stations. In order to diminish the affect of multiple queues being in the optimal phase simultaneously, timeout values that trigger the optimal delay being re-evaluated are picked randomly from a window, and staggered on first startup.

After the optimal discovery phase has completed, the queue's normal operation begins. A queue's normal operation consists of an evaluation period followed by adjustments to the EDCA values based on performance during the evaluation period. The goal of the adjustments is to keep the DF in an acceptable range. Adjustments are made based on comparing the current DF value to the DF\_TARGET for the TC queue. This secondary value is called the DF\_RATIO. Table 3 shows example values that demonstrate how the DF\_RATIO of the current DF to the

DF\_TARGET influences parameter changes. A decrease in priority is defined as a numerical increase in the value of a queue parameter, while an increase in priority is defined as a numerical decrease in the parameter's value.

The distinction between an aggressive change and a relaxed change is made by the order in which the parameters are changed. For example, according to the ordering explored in [4], an aggressive increase in priority would first try adjusting AIFS until exhausted, then try  $CW_{max}$ , then if the other two could not be adjusted,  $CW_{min}$  would be tried. A relaxed adjustment would adjust the parameters in the opposite order. The numerical lower and upper bounds of adjustment are the static TC0 and TC3 parameter sets, respectively. A parameter is considered exhausted when it has been numerically decreased to the TC0 value or numerically increased to the TC3 value. Both AIFS and  $CW_{min}$  are adjusted in units of 1, giving AIFS 6 possible values, and  $CW_{min}$  25 possible values.  $CW_{max}$  is adjusted in units of 64 which reduces the total number of possible values from 1009 ( $15 - 1023$ ) to 16.

Even though only small adjustments are made when a queue is near its DF\_TARGET goal, volatile network conditions can result since the DF\_RATIO will never be equal to the DF\_TARGET, and adjustments will always be required. By design, the algorithm will fluctuate within the window of small adjustments in an attempt to keep the DF as close to the DF\_TARGET (when  $DF\_RATIO = 1.0$ ) as possible. For this reason, a hysteresis mechanism, referred to as tolerance, was added. The tolerance value adds padding to the decision to make parameter adjustments. When the DF has been in the acceptable range for a period of time, meaning the tolerance is high, the DF must be out of the acceptable range multiple times to trigger a change. Inversely, when the DF has been out of the acceptable range for a period of time, meaning the tolerance is low, the DF must be in acceptable range multiple times in order to guarantee that an

adjustment will not be made the next time an unacceptable DF is found. The tolerance mechanism helps the algorithm settle into a steady state by remembering the trend of previous time periods, and reacting more slowly to new trends when the previous period has been stable, and reacting more quickly when the previous period has been volatile.

DF_RATIO	< 0.5	0.5 – 0.8	0.8 – 1.2	1.2 – 2.0	> 2.0
Change in Priority	Aggressive decrease	Relaxed decrease	none	Relaxed increase	Aggressive Increase

Table 2 - Priority Changes

## METHODOLOGY

### Network Topology

The ns2 simulator version 2.28 was used with an EDCA add-on [5] to simulate an 802.11e network with the topology shown in Figure 5. Figure 5 shows the generic structure used in all the simulations. Each wireless node is connected to a single base station which is in turn connected to a wired hub. The hub is then connected to W1, the destination node which is the common destination for all sources. For simplicity, the diagram shows a single source at each station, but a station can have multiple traffic sources. The wireless stations use a data rate of 54 Mbps based on the data rate and physical characteristics of homogenous 802.11g, while the wired stations use 802.3 at 100 Mbps.

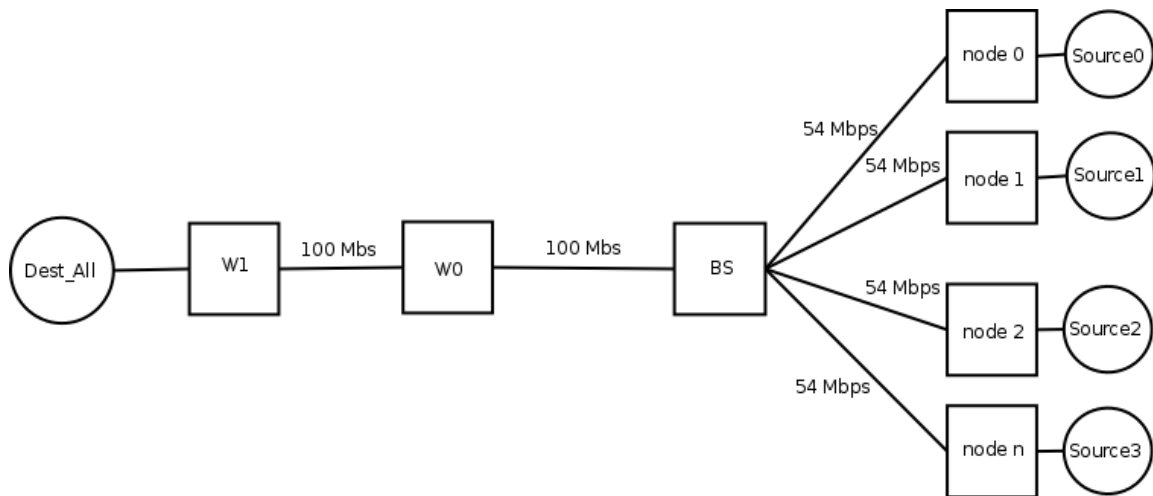


Figure 5 - Simulation Network Topology

### Data Stream Profiles

Table 3 shows the Traffic Class parameter values used in the simulations which have been taken from the 802.11e specification, and are the defaults for the EDCA add-on. TC0 represents the highest priority, while TC3 represents the lowest priority. TC2 values are considered the default values for traffic that is unclassified, therefore for simulations where the Adaptive Algorithm is used, TC2 values are the starting parameter values.  $TxOp_{max}$  was not adjusted dynamically and was disabled for AA simulations, and left default for Static simulations. All AA simulations begin with parameter values set to TC2 values.

Table 4 shows the traffic characteristics of the data flows used in the simulations. The data flows were chosen to provide a mix of traffic types and are based on reasonable voice, video, and data profiles. The profile of G.729 VoIP traffic is modeled with a constant bit rate (CBR) traffic generator configured with a sending rate of 8 Kbps. High priority data is also modeled as TCP/Reno CBR traffic with Delayed ACKs transmitting at a data rate of 13 Kbps, while low priority data has a data rate of 248 Kbps using UDP CBR traffic.

TC / Param	TC0	TC1	TC2	TC3
AIFS	2	2	3	7
$CW_{min}$	7	15	31	31
$CW_{max}$	15	31	1023	1023
$TxOp_{max}$	0.003264	0.006016	0	0

Table 3 – Traffic Class Parameters

SD and HD video profiles used were taken from the H.264/AVC standard [15]. Both profiles used in this study are not intended to specifically represent an exact combination of resolution and quality, but could represent the following profiles: SD video at 4 Mbps could represent Extended



Profile Level 2.1 while HD video could represent High Profile Level 3. These profiles can describe any number of combinations of resolution and frame rate, and are only approximations of video traffic that could be used in these scenarios. Both SD and HD video are VBR traffic sources based on a Pareto distribution. During the on period lasting 5 seconds, the model transmits a burst of data at a configured maximum burst rate, and ceases to transmit during the off period lasting 1 second. The Pareto model's shape parameter was set to 1.4 which results in traffic with frequent bursts.

DF\_TARGET values for each traffic profile were picked based on desired performance and relative performance between TCs. The Voice profile is intended to approximate TC0 performance, and is therefore given a DF\_TARGET of 1.5. A target value of 1.5 means acceptable DF values are values centered around 1.5 times the optimal delay. Since the optimal delay represents an approximation of the best possible performance, a DF\_TARGET near 1.0 will reduce all parameters to TC0 values. Similarly, the video profiles are given DF\_TARGET values of 20.0 which were observed to be values that would consistently result in performance near TC3. The High Priority Data profile was given a DF\_TARGET roughly twice that of Voice, and Low Priority Data was given a DF\_TARGET roughly twice that of High Priority Data.

Type / Attribute	Voice	High Priority Data	Low Priority Data	SD Video	HD Video
Protocol	UDP	TCP/Reno	UDP	UDP	UDP
Profile	CBR	CBR	CBR	VBR	VBR
Rate (kbps)	8	13	248	4000	12000
Packet (bytes)	20	1000	1000	2048	2048
DF_TARGET	1.5	2.5	5.0	20.0	20.0
Normal Priority	TC0	TC1	TC2	TC3	TC3
Metrics	Throughput, Delay, Jitter	Round Trip Time	Throughput, Delay, Jitter	Throughput, Delay, Jitter	Throughput, Delay, Jitter

Table 4 - Traffic Types and Attributes

#### Metrics and Data Collection

The ns2 and EDCA code was extended to allow data collection at the MAC layer. The main metric used to evaluate the performance of the system with the Adaptive Algorithm is the MAC access delay, which is the time from a packet's entrance into the MAC queue until the time it is successfully sent. Arriving packets were time stamped at queue entry and at packet transmission so that the average delays could be calculated. The enqueue event is equivalent to the handing down of a packet from the network layer to the link layer for transmission. The time between the enqueue event and the transmit event is the time spent in the queue, or in other words, is the time a packet is delayed by its contention with other queues and stations. Using the access delays, a mean access delay is accumulated between algorithm execution intervals.

In addition to mean access delay, which is directly used in the algorithm, a number of application specific performance measures were obtained in order to evaluate the network and the successful operation of the Adaptive Algorithm. These measures include end-to-end throughput, transport level delay, jitter, and round trip time (RTT). For each traffic source, there is a

corresponding traffic sink where the received packet is processed. The metrics used for each type of data are shown in Table 4.

Throughput, delay, jitter, and RTT measurements are printed to text files as the simulation runs, and are processed after the simulation has finished. Due to the extreme number of samples produced in each simulation, each raw output file was processed to create per-second average values. The graphs and data shown in the next few sections are average rates per-second of simulation time with units in kilobits/sec and seconds unless otherwise indicated.

For simplicity of tracing, debugging, and graphing each station only includes one data stream for each TC, and therefore a maximum of four data streams per station. Using a setup such as this allows easier tracing of events since the parameters for each TC are guaranteed to be manipulated based on a single performance profile. While this may not be completely realistic, the data streams modeled in this research attempt to encompass a wide variety of data profiles that could be found at a single station.

Before each simulation, the random number generator for the ns2 simulator was seeded with the second value from the current time. Additionally, the start time of each of the data streams in the simulations was randomized within a start-up time window. Although repeatability metrics are not reported, it was observed that the given results were repeatable.

#### Simulation Scenarios

Data was collected from three simulation scenarios: parameter differentiation, burst traffic, and mixed-source traffic. Each scenario was derived from settings previously shown in Table 4. First, a preliminary set of simulations was performed to confirm the work done in [4]. The ordering

of the effectiveness of the 802.11e parameters is central to the performance decisions made in the Adaptive Algorithm. Therefore a few simple simulations were performed to explore the differentiation possible with each parameter. The burst scenario was designed to show the stability of the AA and the ability to adapt to new conditions. Lastly, the mixed-source scenario was designed to show the AA in use in a realistic manner in comparison to normal prioritization, and no prioritization.

In the parameter differentiation scenario, three simulations were performed using six stations, each with a single CBR UDP source with a data rate of 4Mbps. In each simulation, one station's source used the high priority settings shown in Table 3 for a single parameter while the other five stations used the low priority settings for the same parameter. The high priority settings use TC0 values, while the low priority settings use TC3 values. One simulation was performed for each of the three 802.11e parameters.

The second scenario simulated a steady-state network in which a burst source was introduced. This simulation was designed to show the algorithm's ability to hold an uncrowded network in a steady state, accept a new high priority source while balancing priority, and then return to a steady state when the burst was over. A CBR version of the SD traffic source at 7Mbps was used for each Traffic Class similar to the one used in parameter differentiation. TC1, TC2, and TC3 traffic sources began at time 1 and ended at time 300, and a TC0 source began at time 100 and ended at time 200. All other network characteristics were taken from Table 3. For comparison, static prioritization was shown with the same settings.

The final simulation set performed simulated a typical mix of network traffic using all the traffic sources shown in Table 3. Three stations were used each with the following configuration:

- Station 1 – VoIP, Video\_HD
- Station 2 – Data\_Hi, Data\_Lo, Video\_SD
- Station 3 – VoIP, Data\_Lo, Video\_SD

Data was collected using the AA, Static prioritization, and no prioritization. This scenario shows the AA under a realistic workload in an under-utilized network. The scenario was designed to show that the AA performs similarly to Static prioritization, and better than no prioritization even in an under-utilized network.

## RESULTS AND DISCUSSION

### Parameter Differentiation

Figure 6, Figure 7, and Figure 8 show the results of prioritization with AIFS using TC0 AIFS values for the high priority node and TC3 values for all other parameters and data streams. Figure 9, Figure 10, and Figure 11 show the same with  $CW_{min}$ . Figure 12, Figure 13, and Figure 14 show the results for  $CW_{max}$ .

The results in [4] pointed to three important findings: that AIFS was the best parameter for prioritization from the perspective of the high priority data,  $CW_{max}$  was the best parameter from the perspective of the low priority nodes, and  $CW_{min}$  was the best compromise between the two. Table 5 contains the settings and results from these simulations. Throughput results are measured in Kbps, while time results are measured in milliseconds.

	AIFS		$CW_{min}$		$CW_{max}$	
	High	Low	High	Low	High	Low
AIFS	2	7	3	3	3	3
$CW_{min}$	31	31	7	31	7	7
$CW_{max}$	1023	1023	1023	1023	31	1023
Throughput	3820.41	3174.46	3940.14	3321.21	3629.67	2994.25
ThrptTotal	19692.7		20546.2		18600.9	
ThrptDiff	645.95		618.93		635.42	
Jitter	1.238	3.778	1.149	3.535	3.371	5.819
Delay	6.302	450.224	6.284	412.768	30.427	389.391

Table 5 – Parameter Differentiation

Although small differences can be observed in the graphs, the raw averages from Table 5 provide more direct insight. Contrary to previous findings,  $CW_{min}$  gave better overall network utilization as is evidenced by the total throughput and a smaller difference between the two

classes. Additionally,  $CW_{\min}$  provided lower jitter and delay than AIFS for both low priority and high priority data. It was found that  $CW_{\max}$  was indeed better for low priority nodes due to less aggressive prioritization of the high priority data.

$CW_{\max}$  adjustment provides the least desirable performance due to the fact that it does not come into play until after AIFS and  $CW_{\min}$  in the back-off algorithm of 802.11e. The first action a queue takes when it wishes to transmit is to wait an AIFS. Therefore, small AIFS intervals affect every transmission including successful transmissions, failed transmissions, and collisions. As the CW grows large, the AIFS has less impact on transmission priority, but still plays the most active roll of the three parameters. Small  $CW_{\max}$  values only benefit a queue when there have been multiple collisions and the CW has increased, wherein and the CW can be potentially very large. A queue with a small  $CW_{\max}$  setting will usually have a smaller CW since the queue will reach the maximum value more quickly. Despite giving priority to some nodes,  $CW_{\max}$  values being small also contributes to increased contention in the network since back-off windows grow to smaller sizes which results in increased probability that two stations will have zero back-off counters simultaneously. Smaller values also mean that in a crowded network, retry limits will be reached more quickly, increasing failures and retransmits. For these reasons,  $CW_{\max}$  is least harsh for the low priority data, least beneficial to the high priority data, and most harmful to the network.

$CW_{\min}$  values control the beginning value of the CW, which is the upper bound of the random window from which the back-off counter is chosen. After a queue waits an AIFS and senses the medium to be free, it decrements its back-off counter which begins set to a random value between 0 and CW.  $CW_{\min}$  can be viewed as the second value to be examined in the back-off algorithm, just like AIFS is the first. A queue with a smaller  $CW_{\min}$  value is likely to be able to transmit before

one with a larger value since it is more likely to have a smaller back-off counter. Being able to transmit first means it is less likely to need to double the CW and count down again.

It is possible the difference between the results presented here and those presented in [4] is a result of different values used in the simulations, although the methods were similar. The parameter settings used here are taken from the 802.11e specification that was not finalized at the time of publication in [4]. Despite the contrary results, AIFS was still chosen as the primary parameter for adjustment in the Adaptive Algorithm since it is the primary parameter used by all data as discussed previously.

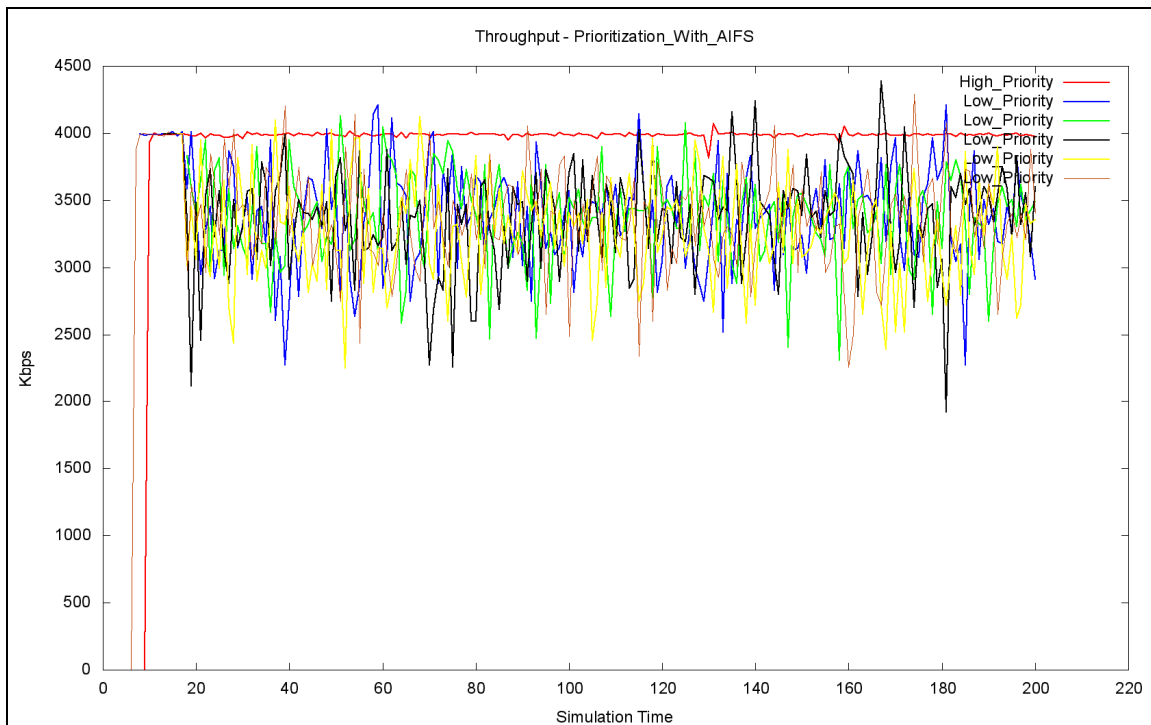


Figure 6 – Throughput, Prioritization With AIFS



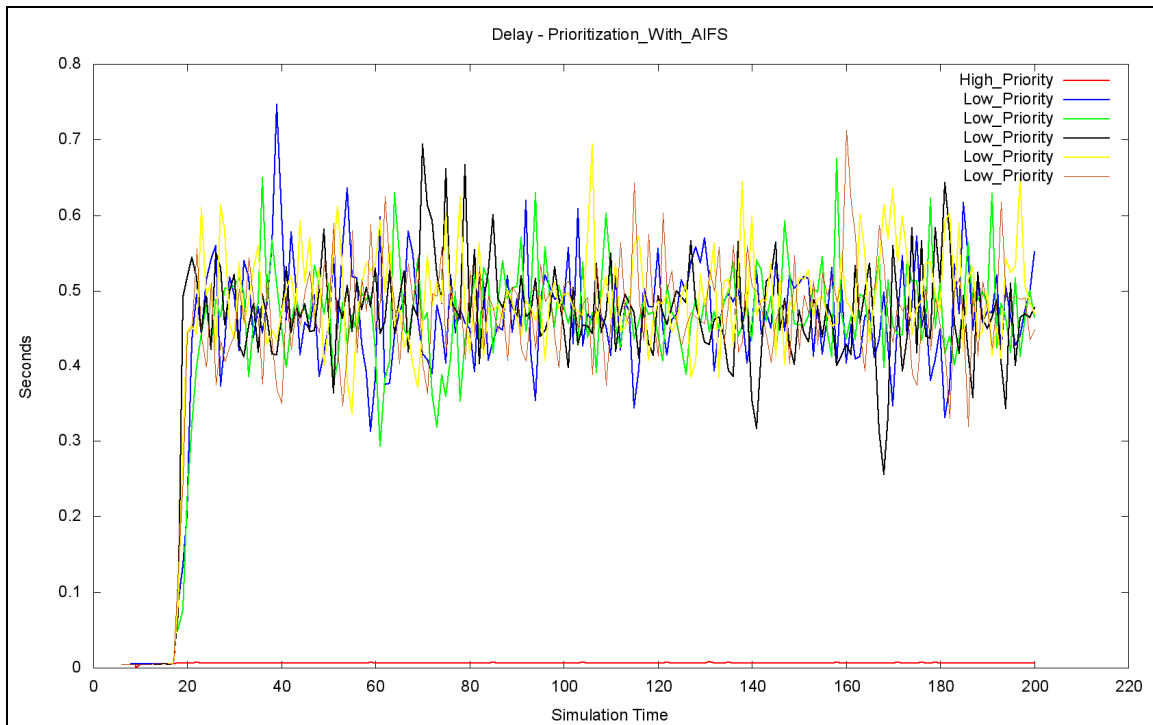


Figure 7 – Delay, Prioritization With AIFS

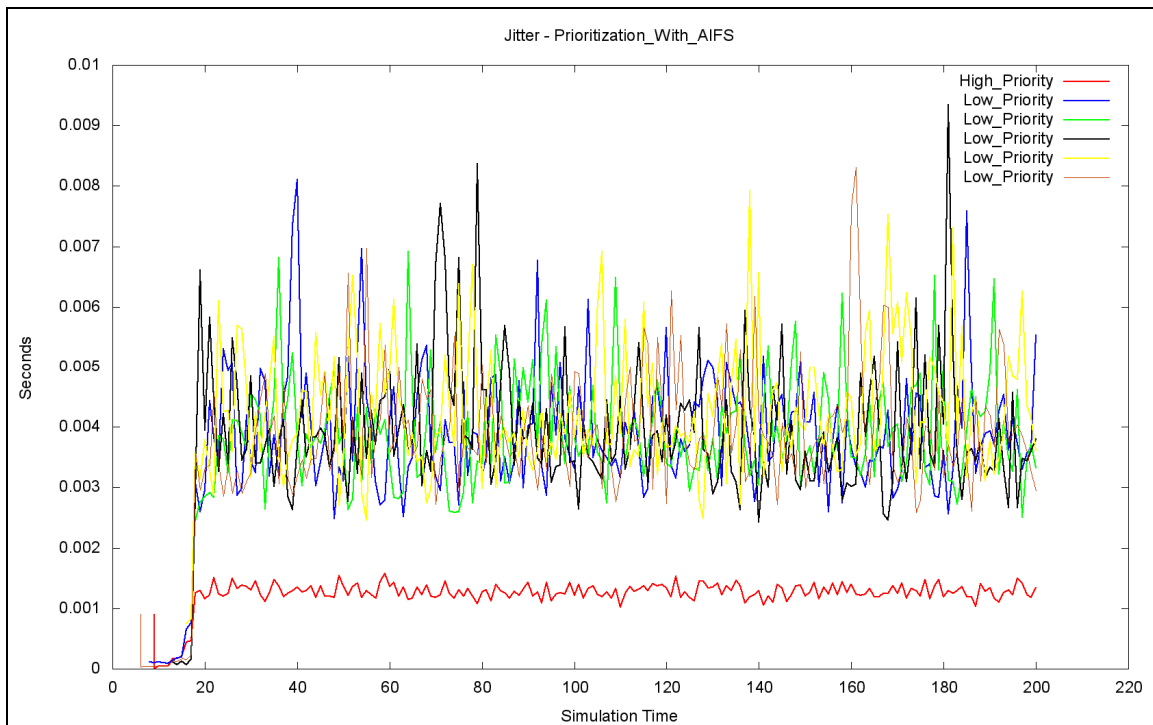


Figure 8 – Jitter, Prioritization With AIFS

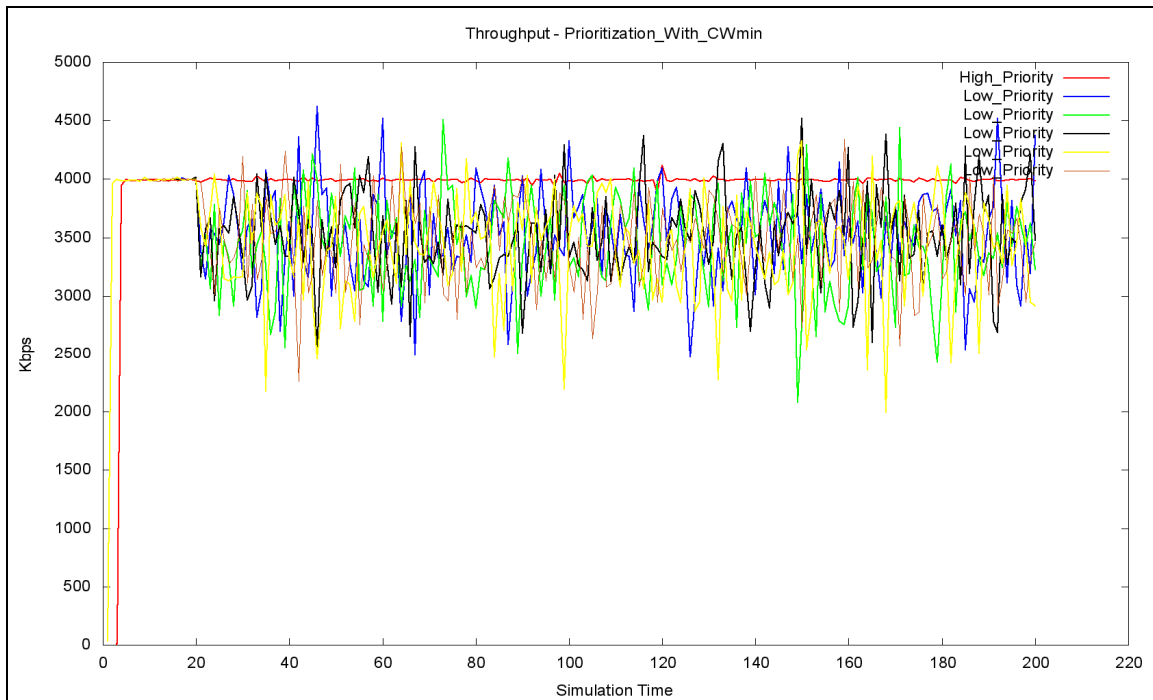


Figure 9 – Throughput, Prioritization With  $CW_{min}$

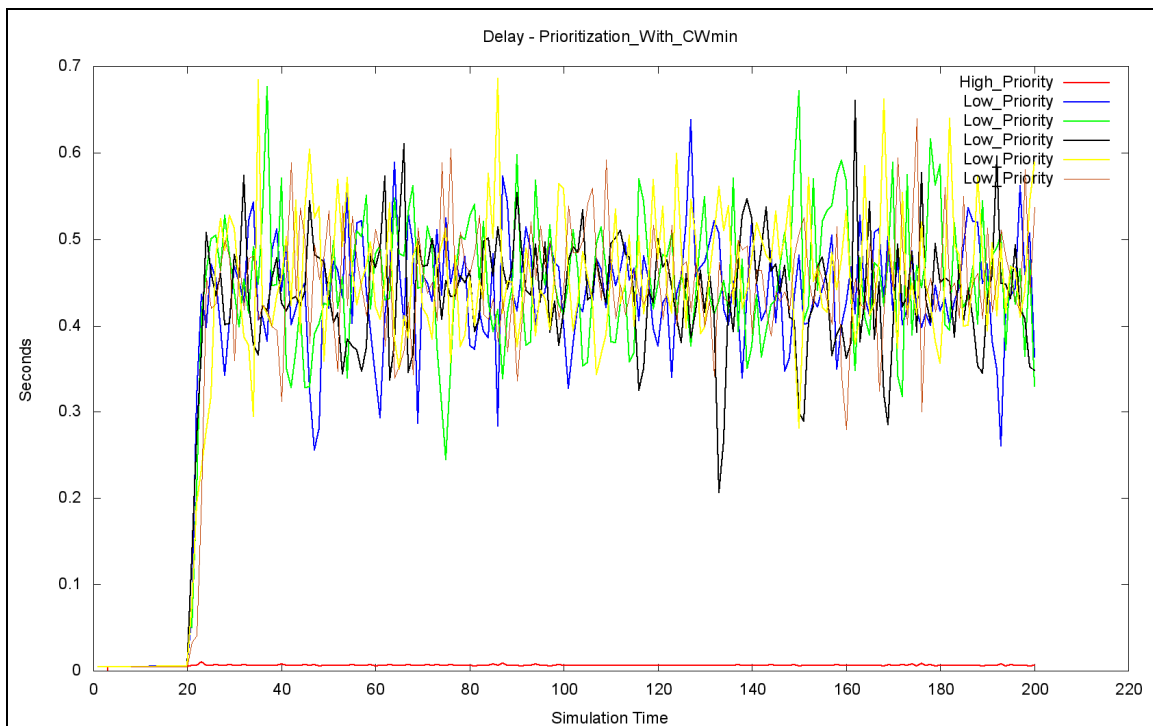


Figure 10 – Delay, Prioritization With  $CW_{min}$

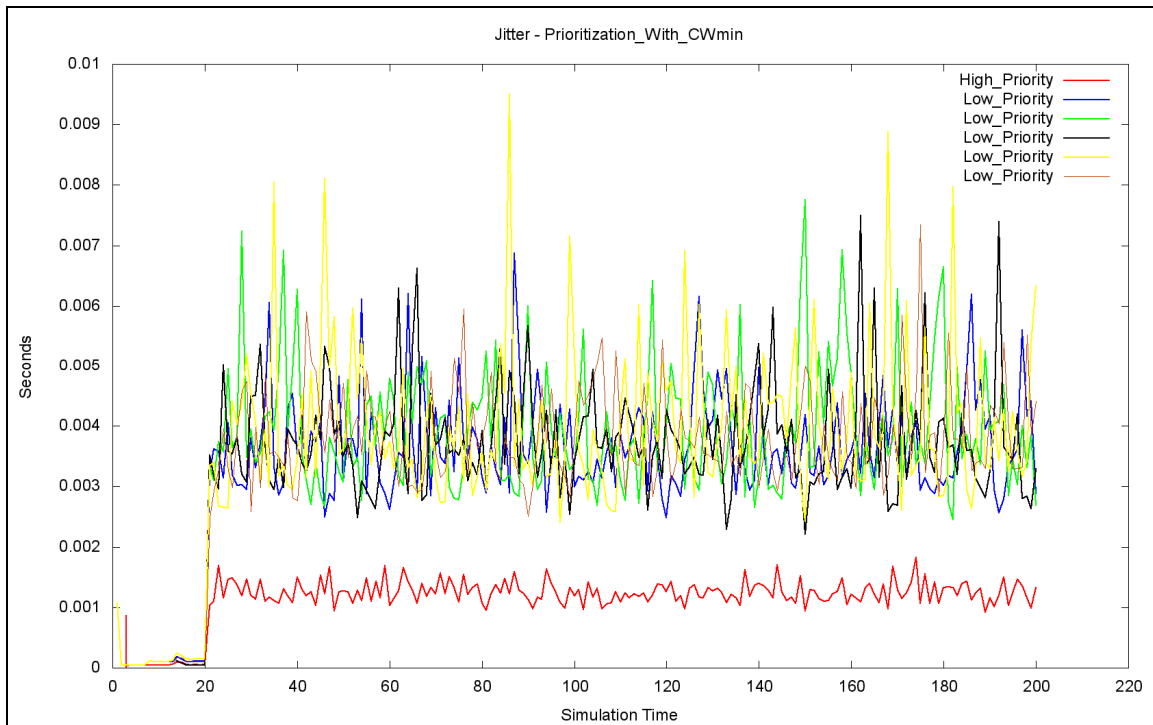


Figure 11 – Jitter, Prioritization With  $CW_{min}$

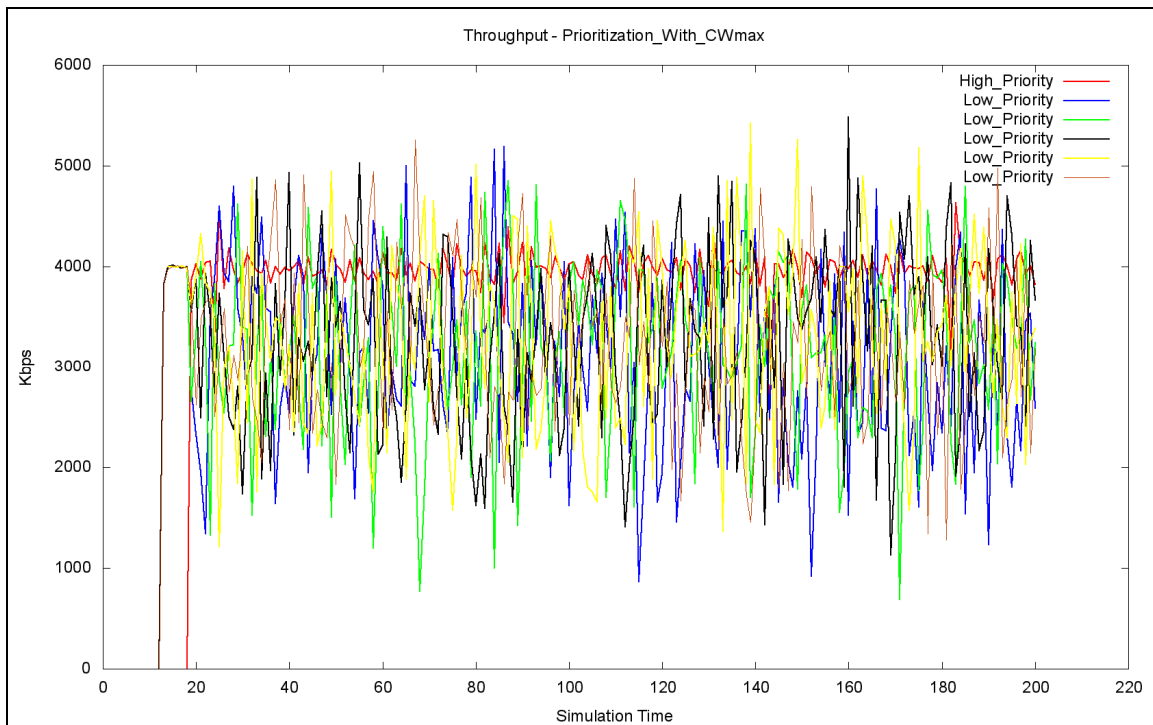


Figure 12 – Throughput, Prioritization With  $CW_{max}$

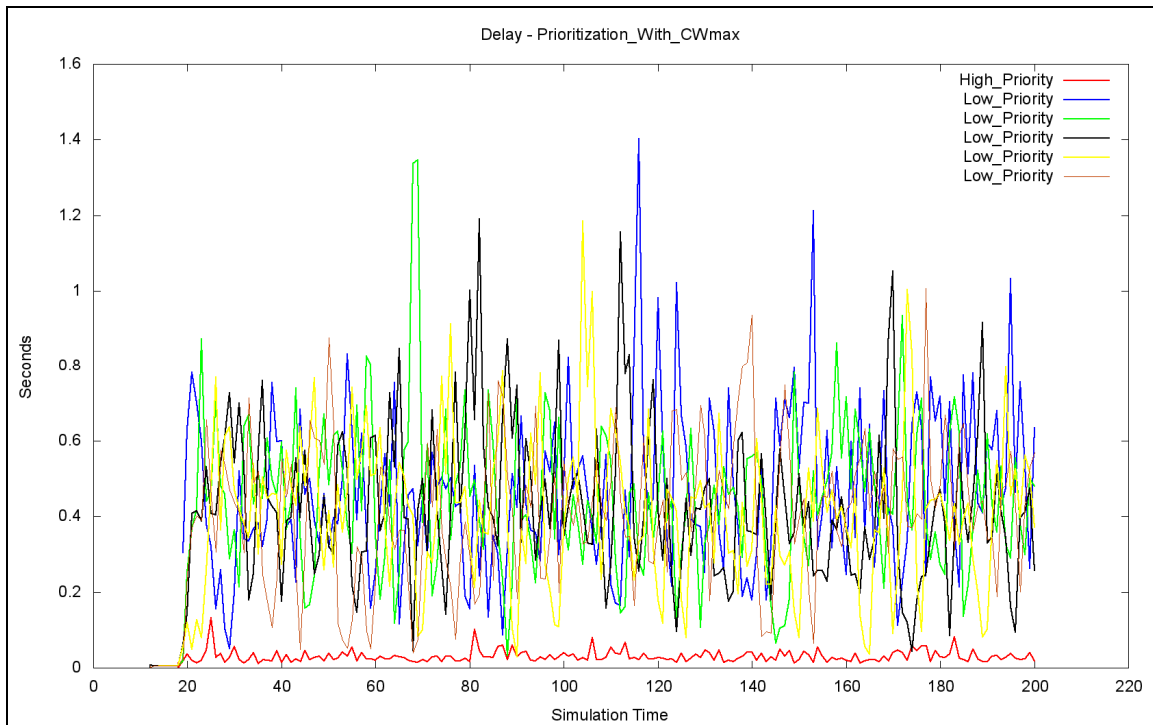


Figure 13 – Delay, Prioritization With  $CW_{max}$

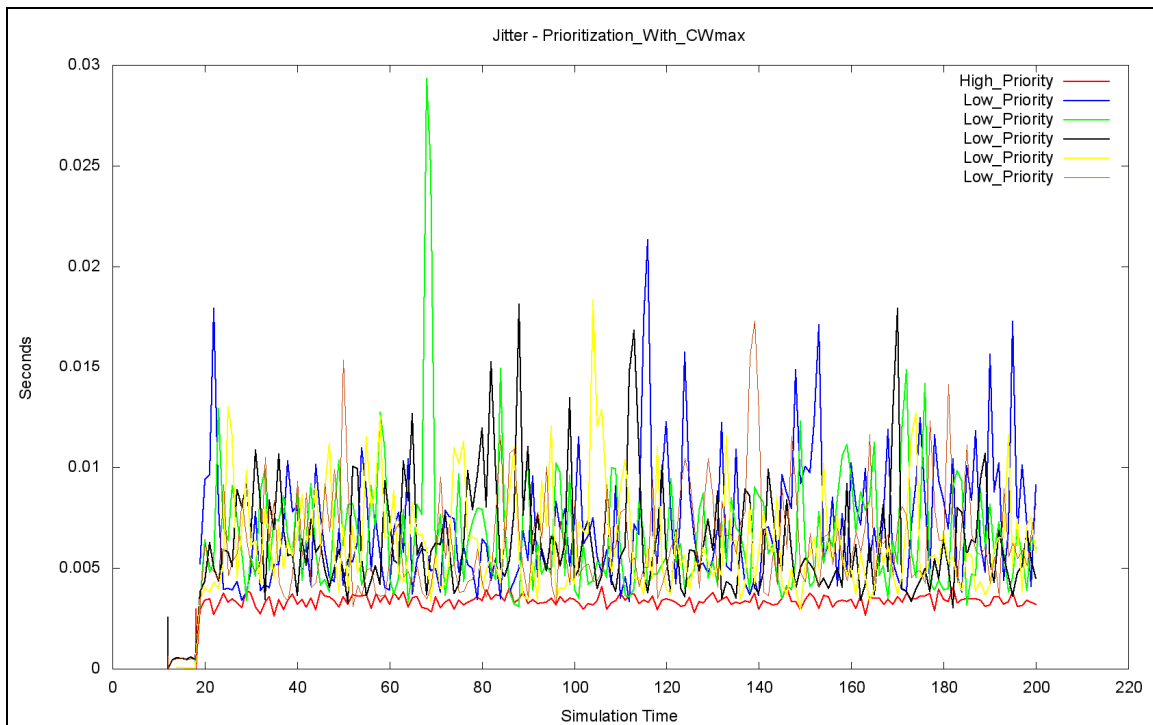


Figure 14 – Jitter, Prioritization With  $CW_{max}$

### Burst Simulations

In the next set of simulations, a burst scenario was simulated in order to show the performance of the algorithm and its ability to adapt to new traffic sources. Figure 15, Figure 16, and Figure 17 show prioritization with the static 802.11e parameters. Figure 18, Figure 19, and Figure 20 show prioritization with the AA. The graphs are summarized in Table 6 and Table 7. The first table shows data from the burst period between simulation time 100 and 200, while the second shows the data from the non-burst period between simulation time 200 and 300. The time period from 0 to 100 is essentially identical to the later third and is not shown.

Traffic Class	TC0		TC1		TC2		TC3	
Prioritization	AA	Static	AA	Static	AA	Static	AA	Static
Throughput	6984.93	6953.28	6999.97	6790.63	4143.51	5499.1	3984.92	2699.13
Delay	12.634	5.461	6.296	6.07	380.696	285.19	396.843	537.559
Jitter	0.583	0.583	0.799	0.88	3.28	1.735	3.53	6.776

Table 6 - Burst Scenario Data

Traffic Class	TC1		TC2		TC3	
Prioritization	AA	Static	AA	Static	AA	Static
Throughput	6999.99	6860.4	6986	6627.85	6981.29	6906.9
Delay	5.171	5.163	15.5	4.968	20.475	5.601
Jitter	0.208	0.113	0.408	0.137	0.4	0.163

Table 7 - Non-Burst Scenario Data

The data from the burst period shows that the AA performs similar to Static prioritization with differences in a few interesting areas. During the non-burst period the AA network has an average total throughput of 20,967 Kbps while the Static network has a total of 20,395, and during the burst period the totals are 22,113 Kbps and 21,942 Kbps for AA and Static, respectively. During the non-burst period the total requested bitrates of the data streams is 21 Mbps, and during the burst period the total of the bitrates is 28 Mbps. During the non-burst period the data shows that

since the network is not overloaded, and adjustments are not needed, the AA is able to achieve 3% more efficient network utilization.

Despite more efficient channel utilization, the AA does not deliver as low delay or jitter as the Static method. Since the algorithm's lowest queue parameters allowed are TC0 values, the delay and jitter values of the Static method can be considered a lower bound for the performance the AA can give, therefore the AA will never be able to deliver guarantees as low as the Static method. For real scenarios, this is not a serious problem since over time the difference becomes more negligible, but during short simulations such as these the difference is more pronounced.

Figure 18, Figure 19, and Figure 20 show a disadvantage of the AA compared to Static when adjusting to new network conditions. When network conditions change, such as when the TC0 data stream is introduced, prioritization is slowed by a "reaction time" that is equal the time required to measure a new optimal delay, and to correct parameters based on the new measurements. It is possible to minimize these reaction periods by shortening the interval between optimal delay sampling, but at the risk of making the network more volatile. Naturally, Static prioritization does not suffer from this since the parameters are constant.

In order to explore the ramifications of the new queue structure of 802.11e, an alternative dataset was collected where all four TC data streams resided at the same station rather than being located at four separate stations. The results of this simulation are shown in Figure 21, Figure 22, and Figure 23. In these figures better separation is achieved between each TC, with similar average performance to the multi-node data. The data streams in the single node are able to balance themselves more efficiently than the multi-node data streams since there is only a single level of contention. Since a queue must contend against other queues in its own station,

and against other stations, the absence of other stations reduces contention as well as reducing the affect of other stations on the observed access delays.

It was also observed, although not shown, that the internal queue structure within the station affects prioritization in another important way. Referring to Figure 3, when the back-off counters of two queues reach zero simultaneously, the queue with the lower queue number ( $TC0 < TC1 < TC2 < TC3$ ) is given the chance to transmit and the queue with the higher value experiences a virtual collision. Therefore in a single node, multi-priority scenario such as shown in Figure 21, Figure 22, and Figure 23, even when DF\_TARGET values are set to maximum amounts where each queue will quickly adjust its parameters to TC3 values and remain there, prioritization occurs. In other words, simulations using the Single Node settings with equal DF\_TARGET values results in prioritization with separation similar to that shown in the three Single Node figures.

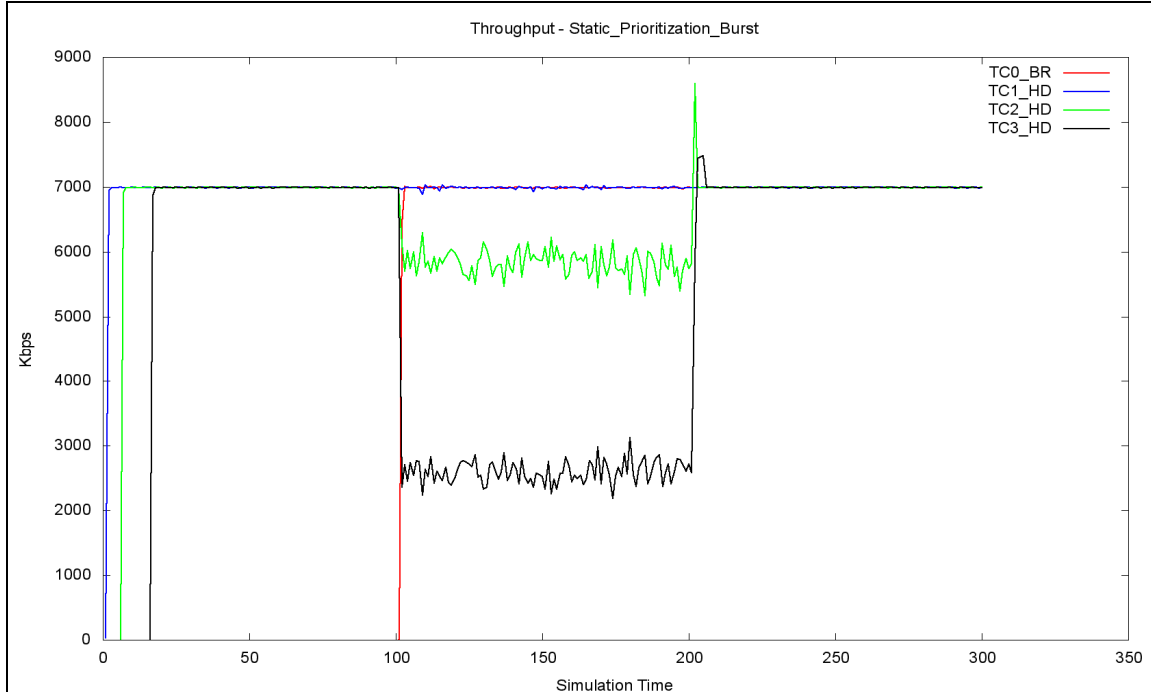


Figure 15 - Throughput, Static Prioritization With Burst

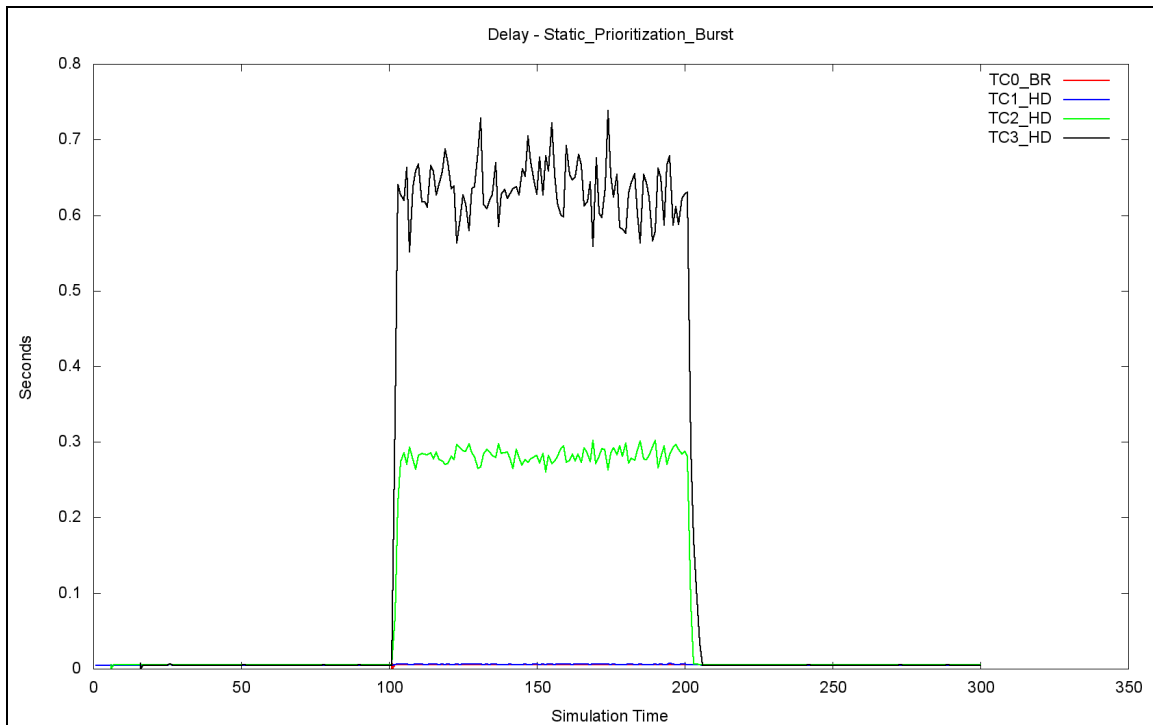


Figure 16 - Delay, Static Prioritization With Burst

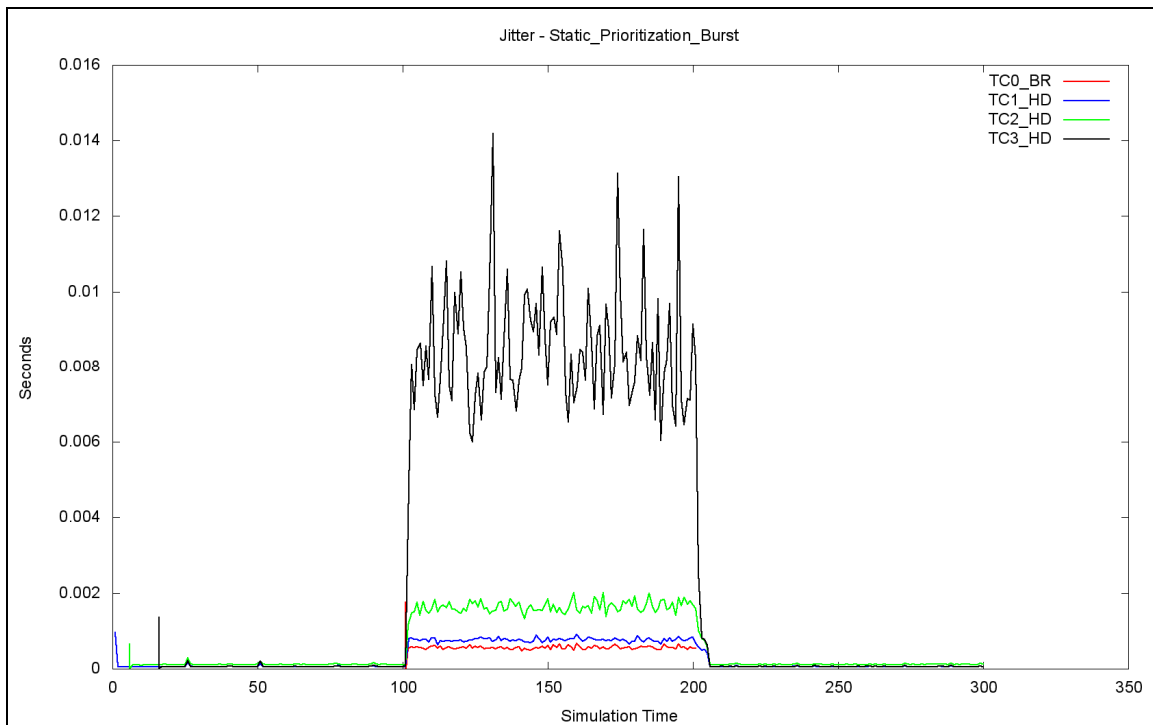


Figure 17 - Jitter, Static Prioritization With Burst



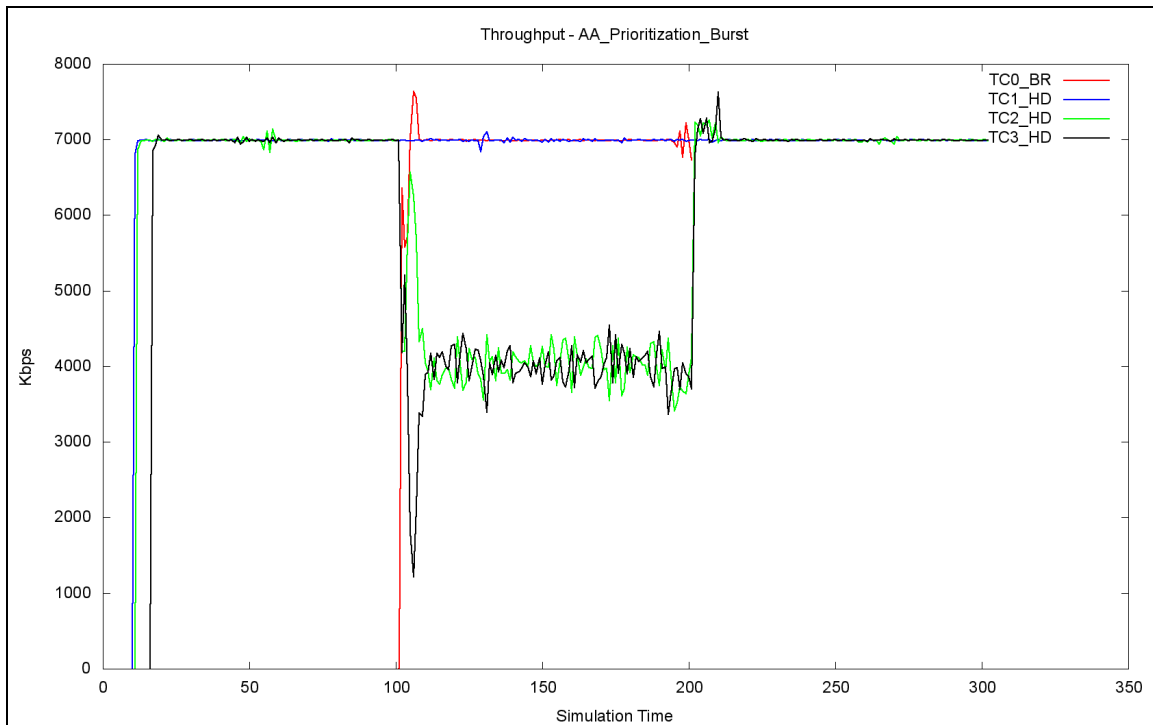


Figure 18 - Throughput, AA Prioritization With Burst

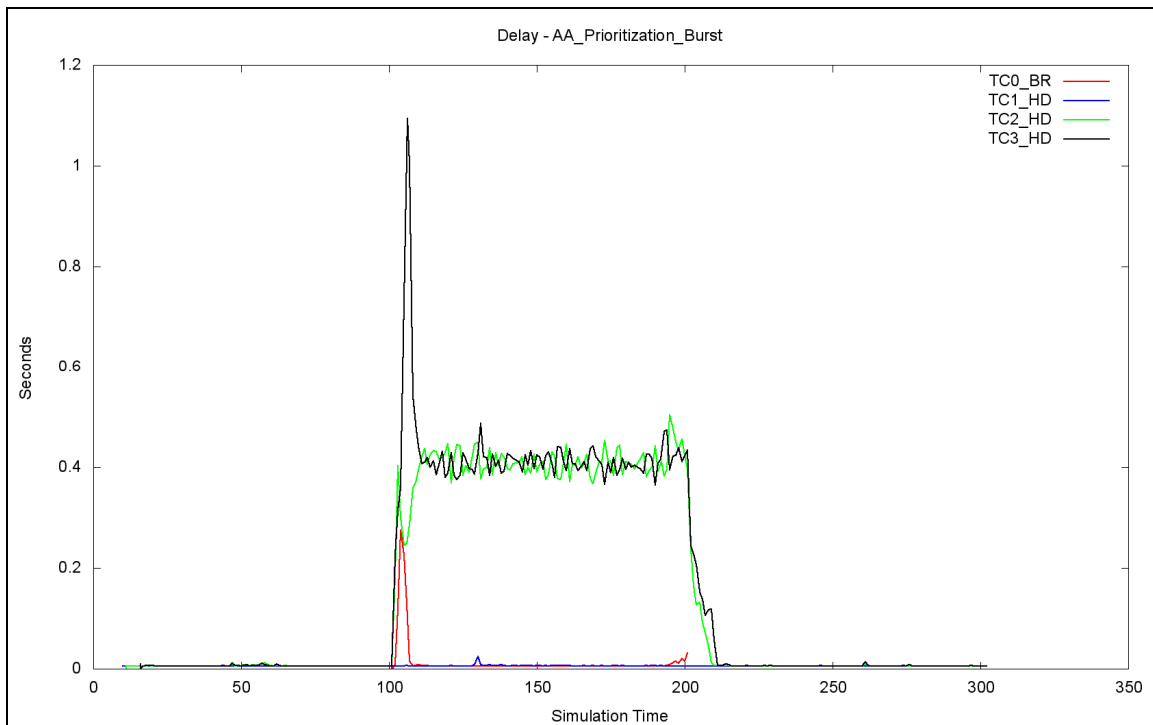


Figure 19 - Delay, AA Prioritization With Burst

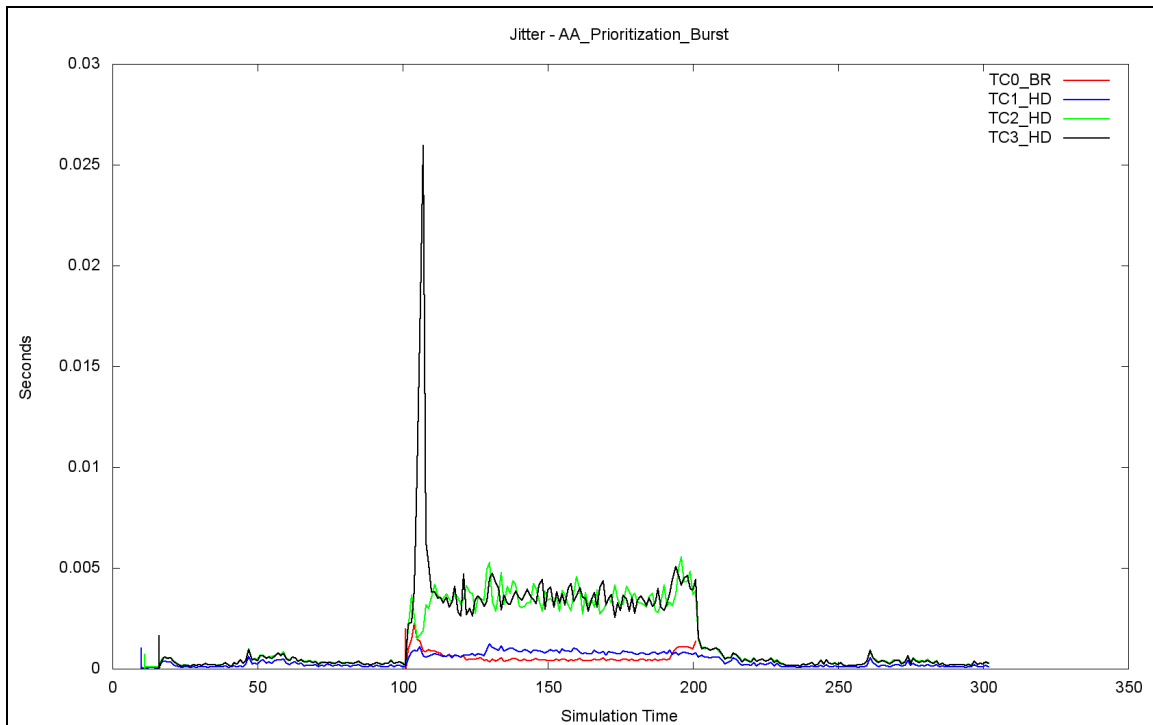


Figure 20 - Jitter, AA Prioritization With Burst

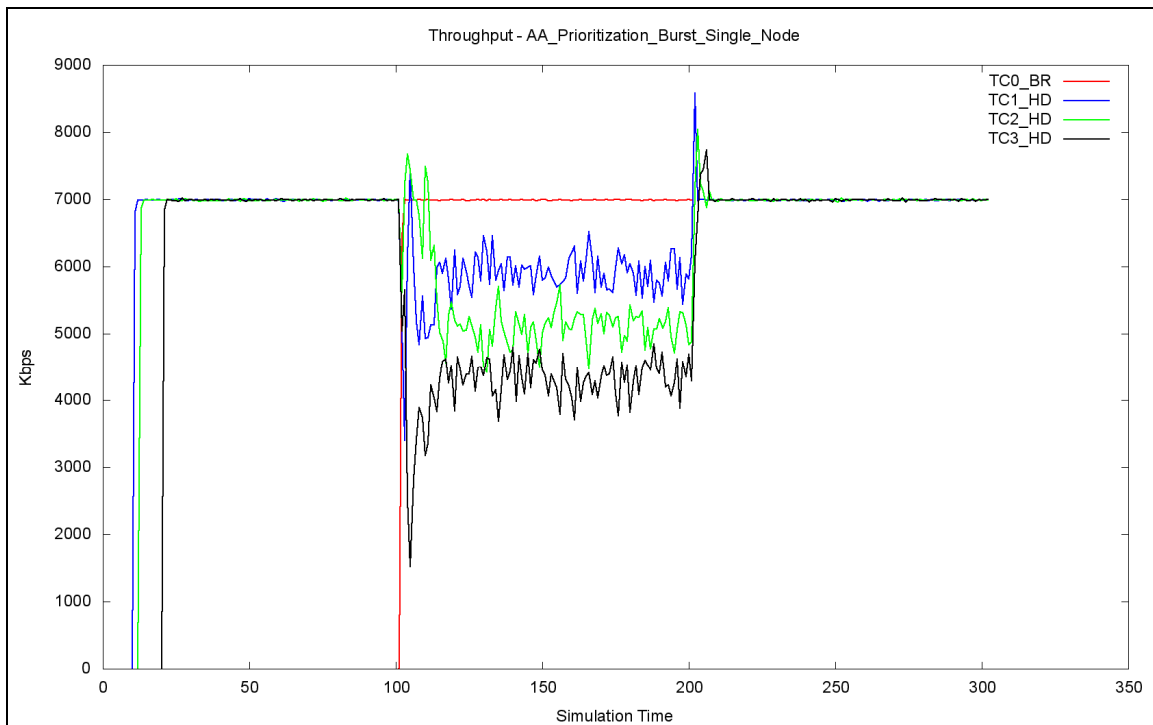


Figure 21 - Throughput, AA Prioritization With Burst, Single Node

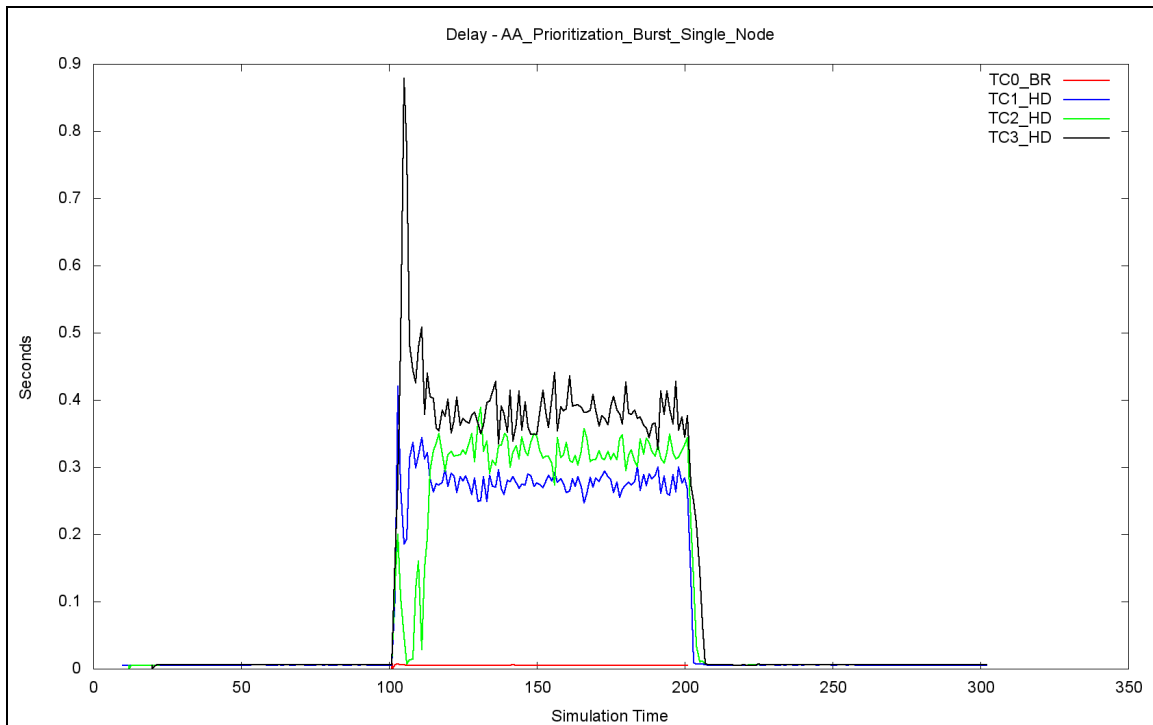


Figure 22 - Delay, AA Prioritization With Burst, Single Node

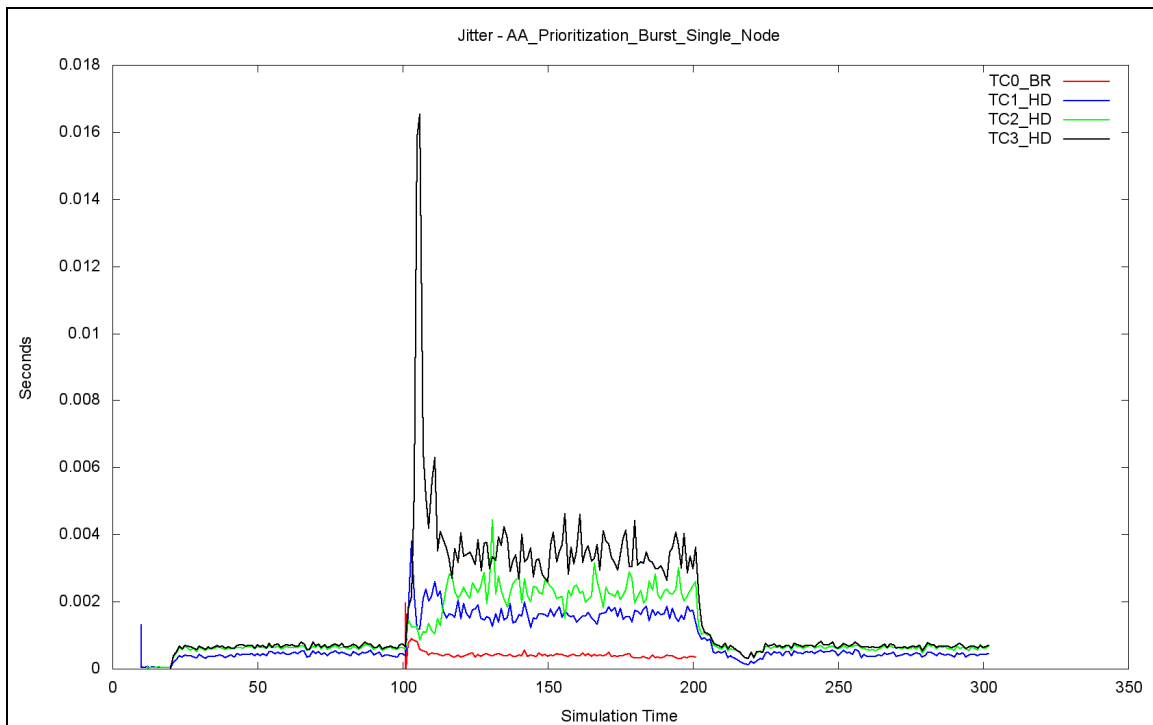


Figure 23 - Jitter, AA Prioritization With Burst, Single Node

### Mixed Traffic Simulation

Figure 24, Figure 25, and Figure 26 show the results of prioritization with the Static method in the mixed data environment. Figure 27 and Figure 28 show adjusted throughput and delay graphs with video sources removed so that lower values are more easily visible. Table 9 shows the summary of these figures. Prioritization with the AA is shown in Figure 30, Figure 31, and Figure 32. Range-adjusted graphs for AA are shown in Figure 34 and Figure 35. A summary of the AA is shown in Table 8. For baseline reference, the same simulation with no prioritization is shown in Table 10.

Data Type	VoIP	Data_Lo	Data_Hi	Video_SD	Video_HD
Throughput	8.00	248.00	13.86	3210.08	10206.34
Delay/RTT	4.768	5.826	114.23	6.919	40.26
Jitter	0.445	1.27		1.088	0.52

Table 8 - Data Summary, AA Prioritization, Mixed

Data Type	VoIP	Data_Lo	Data_Hi	Video_SD	Video_HD
Throughput	8.00	248.00	13.86	3480.36	10499.22
Delay/RTT	4.755	5.358	106.88	6.176	83.891
Jitter	0.426	0.814		1.08	0.598

Table 9 - Data Summary, Static Prioritization, Mixed

Data Type	VoIP	Data_Lo	Data_Hi	Video_SD	Video_HD
Throughput	8.00	248.00	13.86	3326.02	9028.46
Delay/RTT	5.424	5.422	113.08	5.794	20.088
Jitter	1.063	0.95		0.801	0.485

Table 10 - Data Summary, No Prioritization, Mixed

The graph data show similar patterns between AA and Static prioritization. However, referring to the tables, more subtle differences can be observed. VoIP shows nearly identical prioritization under Static and AA prioritization, and greatly increased delay and jitter compared to the baseline.

The high bandwidth low priority video data performs best with Static prioritization, although AA is still able to out-perform the baseline with better overall video performance.

In this scenario, the network was not overloaded, and prioritization was less effective since the low priority nodes did not have trouble gaining access to the medium. It is surprising that the AA is less efficient at using the channel than Static prioritization in this scenario. This effect is mostly likely due to the use of VBR video traffic, and its effect on the stability of the algorithm. The AA is able to adapt to different network configurations, but is not able to adapt to wildly fluctuating data streams.

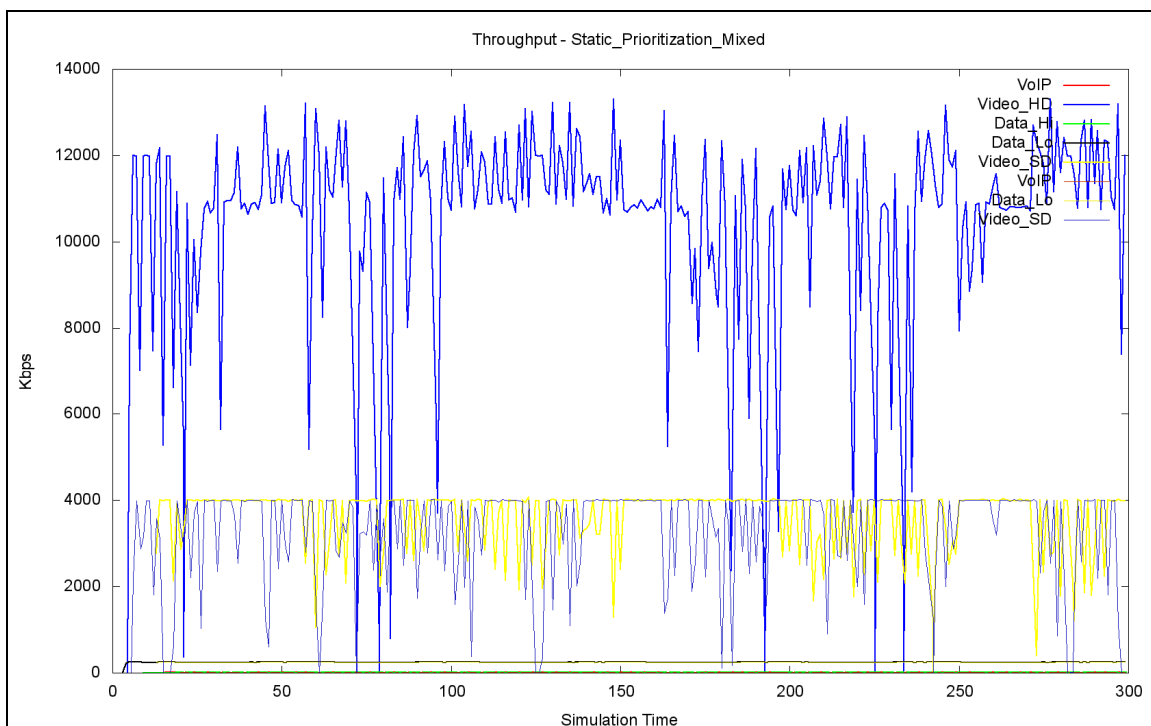


Figure 24 - Throughput, Static Prioritization, Mixed

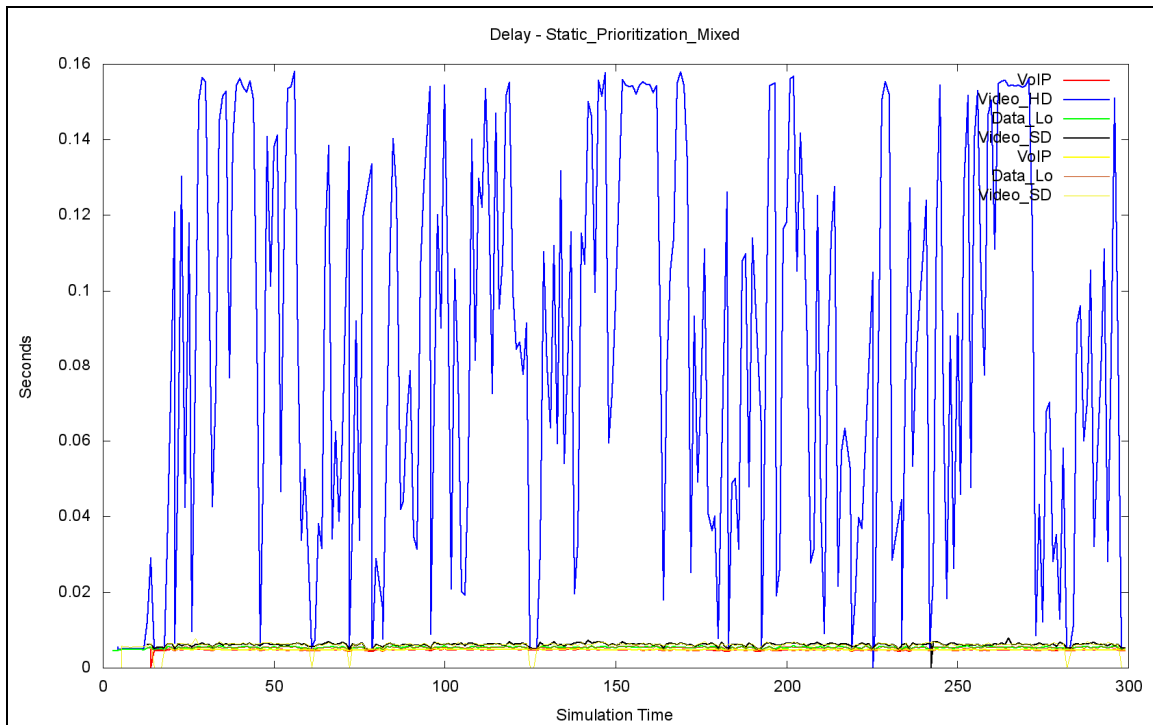


Figure 25 - Delay, Static Prioritization, Mixed

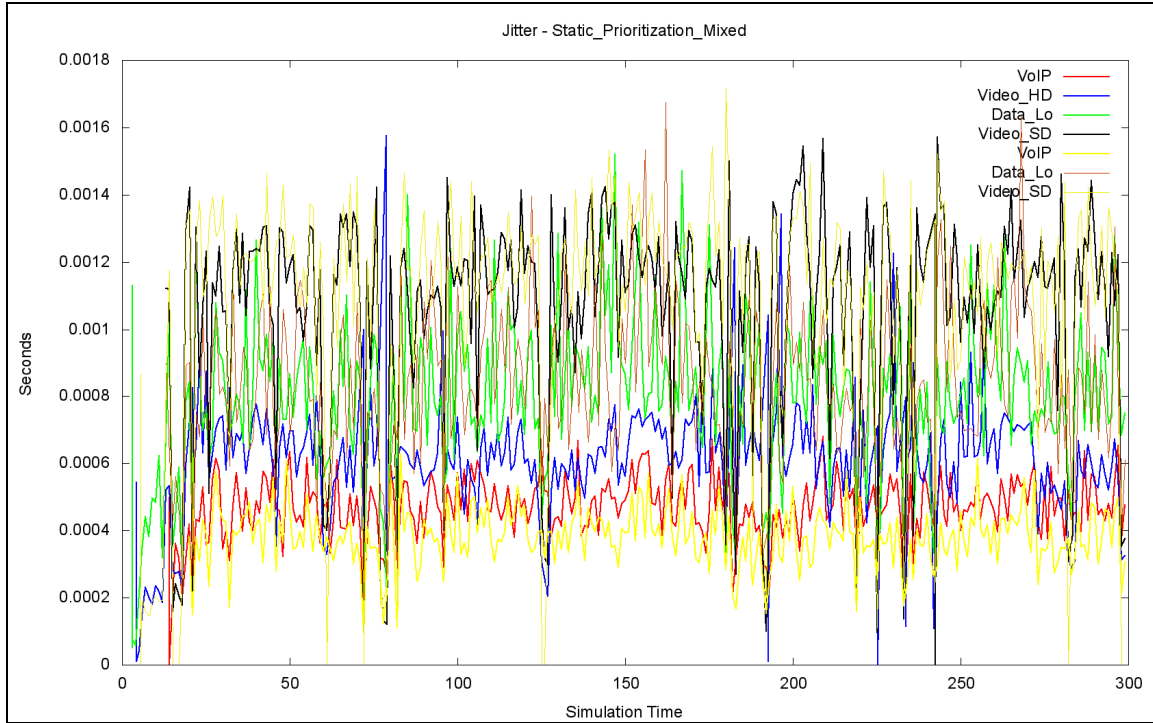


Figure 26 - Jitter, Static Prioritization, Mixed

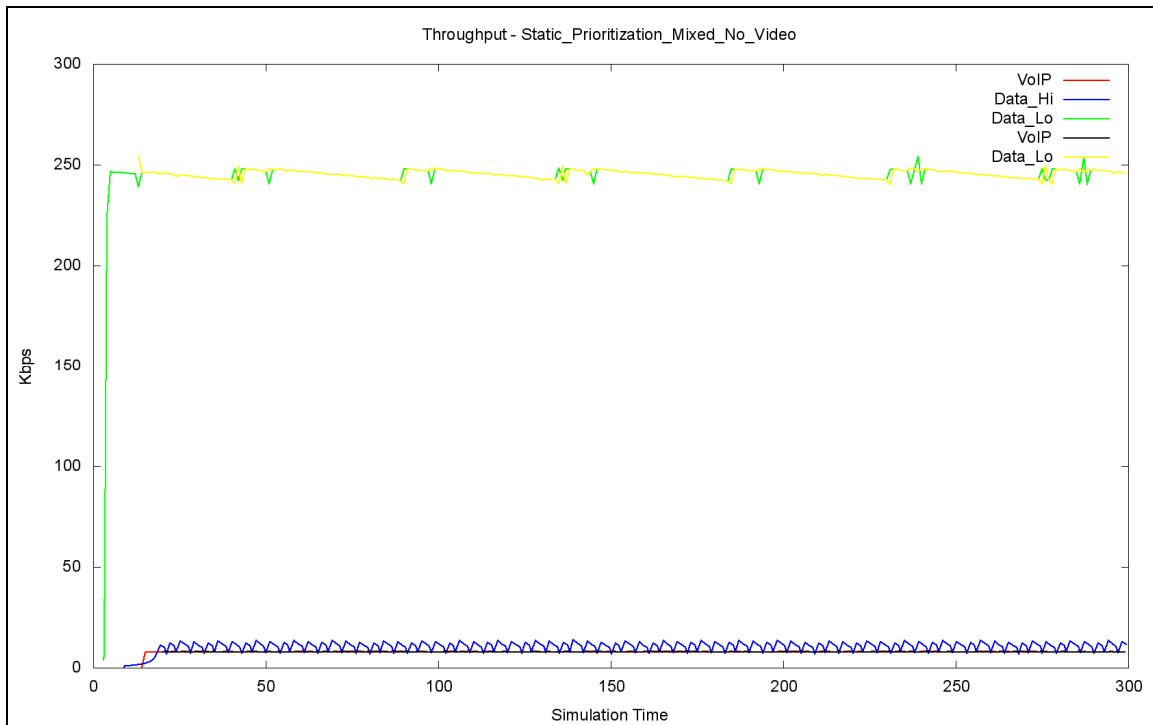


Figure 27 - Throughput, Static Prioritization, Mixed, No Video

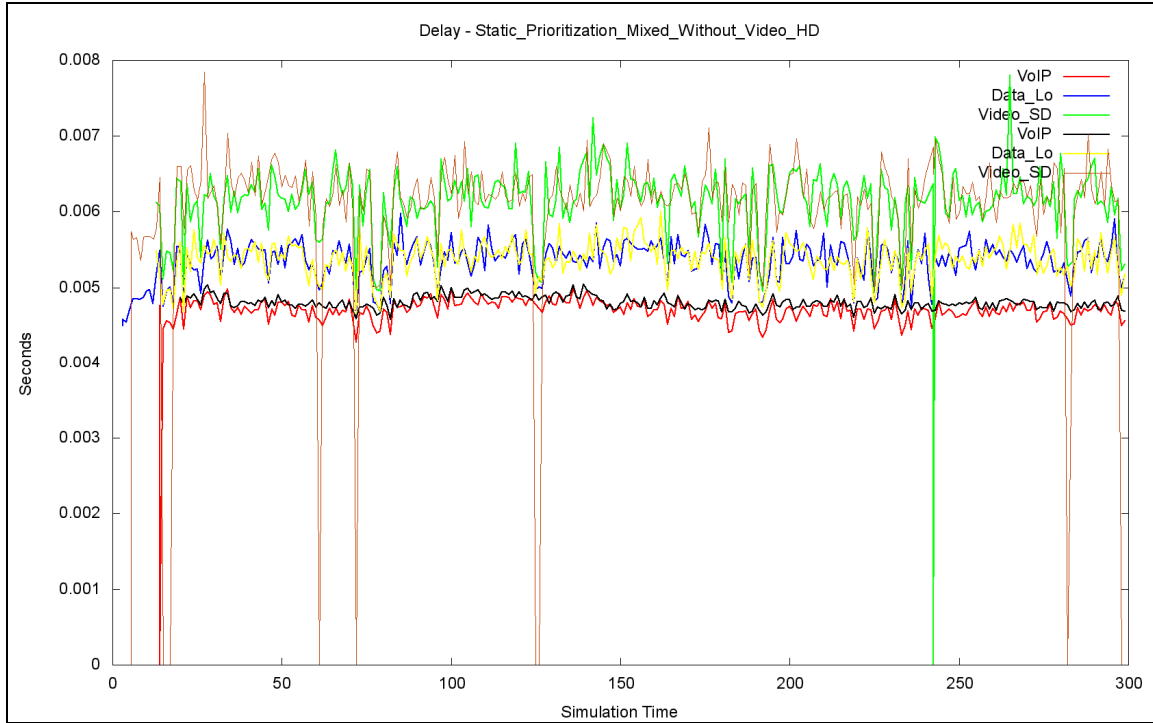


Figure 28 - Delay, Static Prioritization, Mixed, No Video\_HD

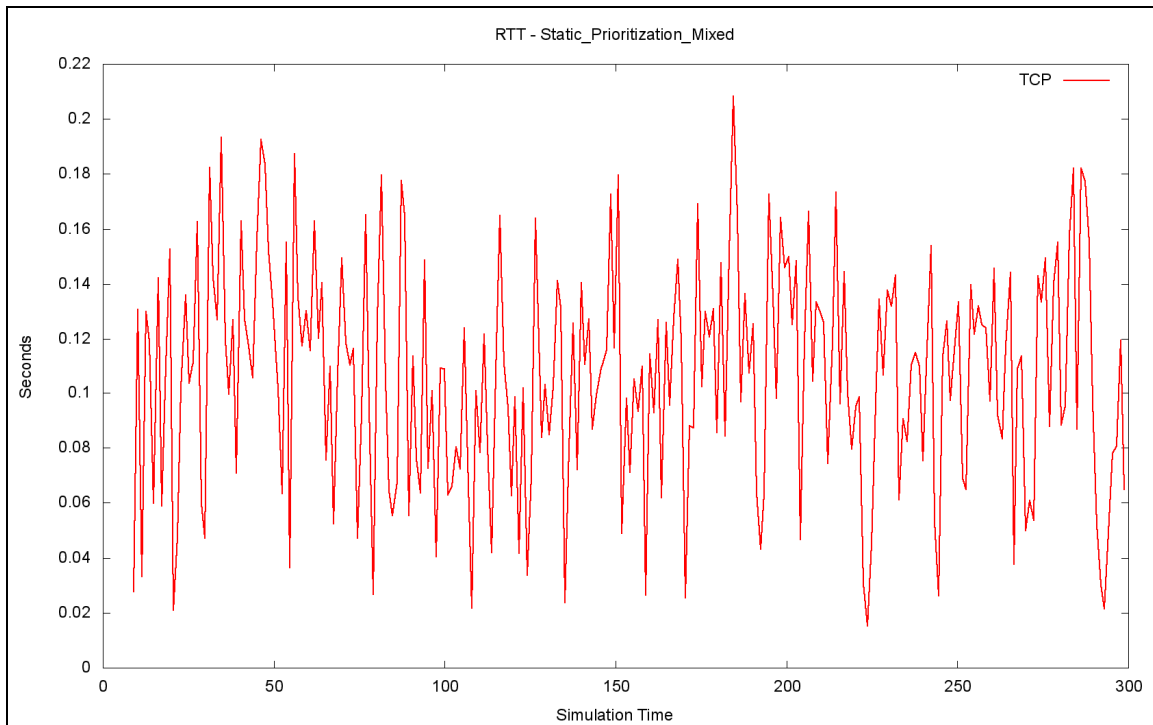


Figure 29 - Round Trip Time, Static Prioritization, Mixed

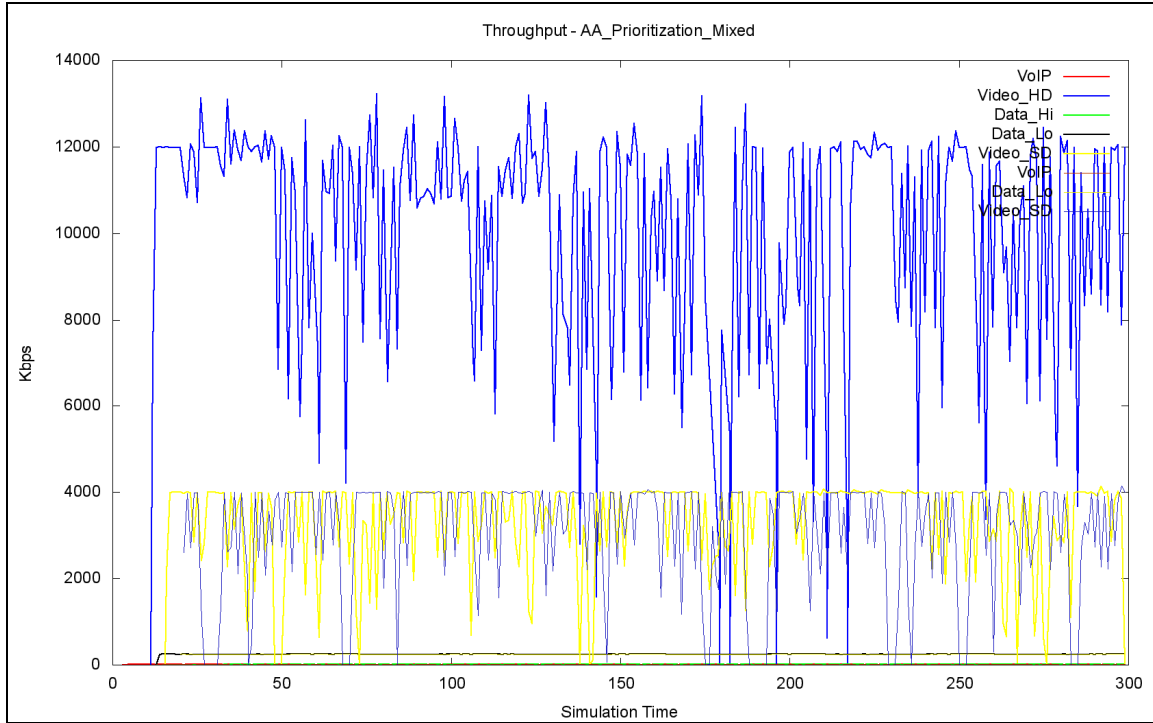


Figure 30 - Throughput, AA Prioritization, Mixed



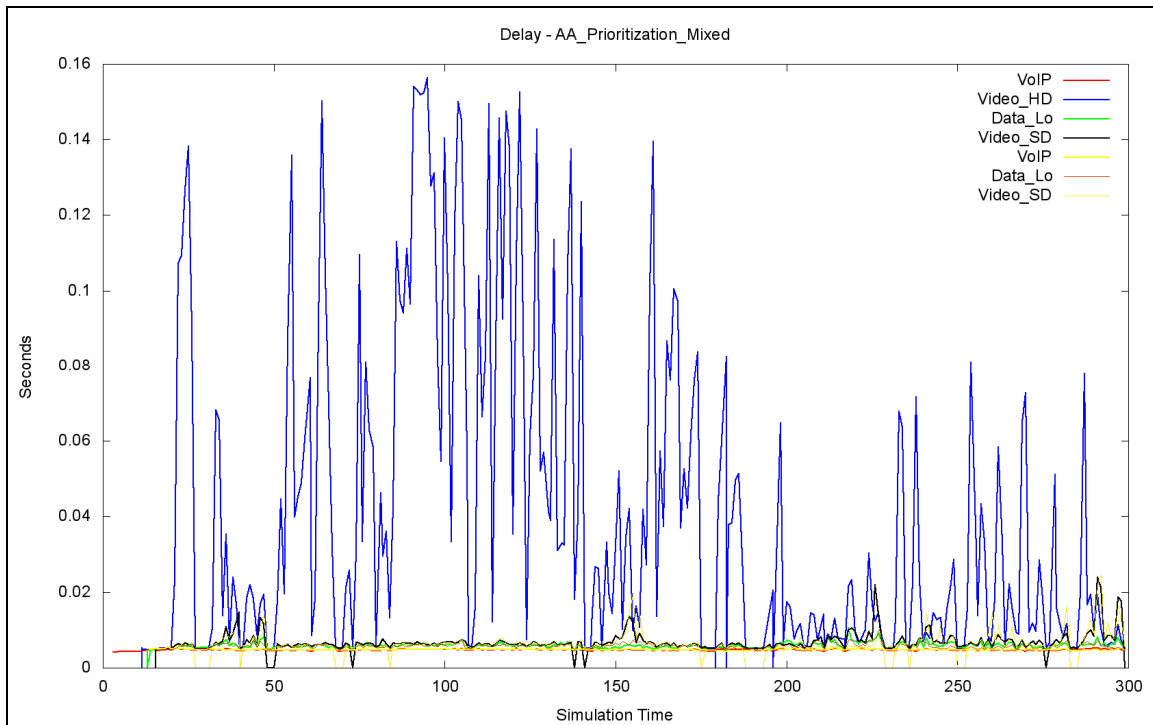


Figure 31 - Delay, AA Prioritization, Mixed

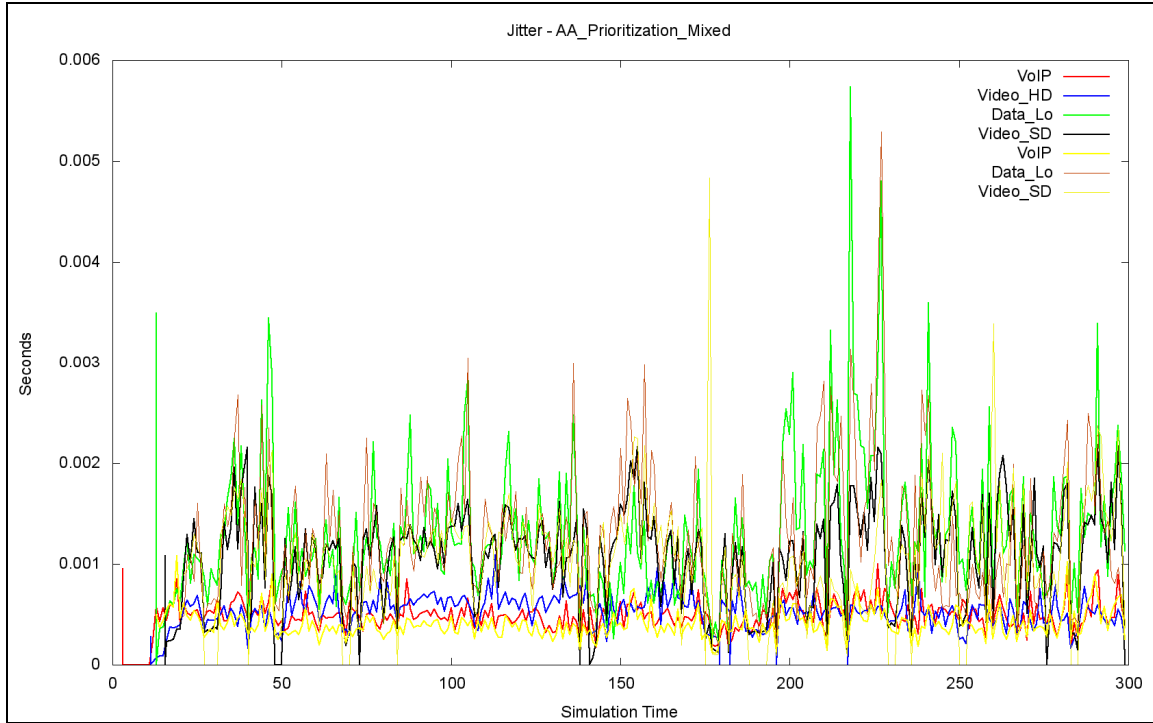


Figure 32 - Jitter, AA Prioritization, Mixed

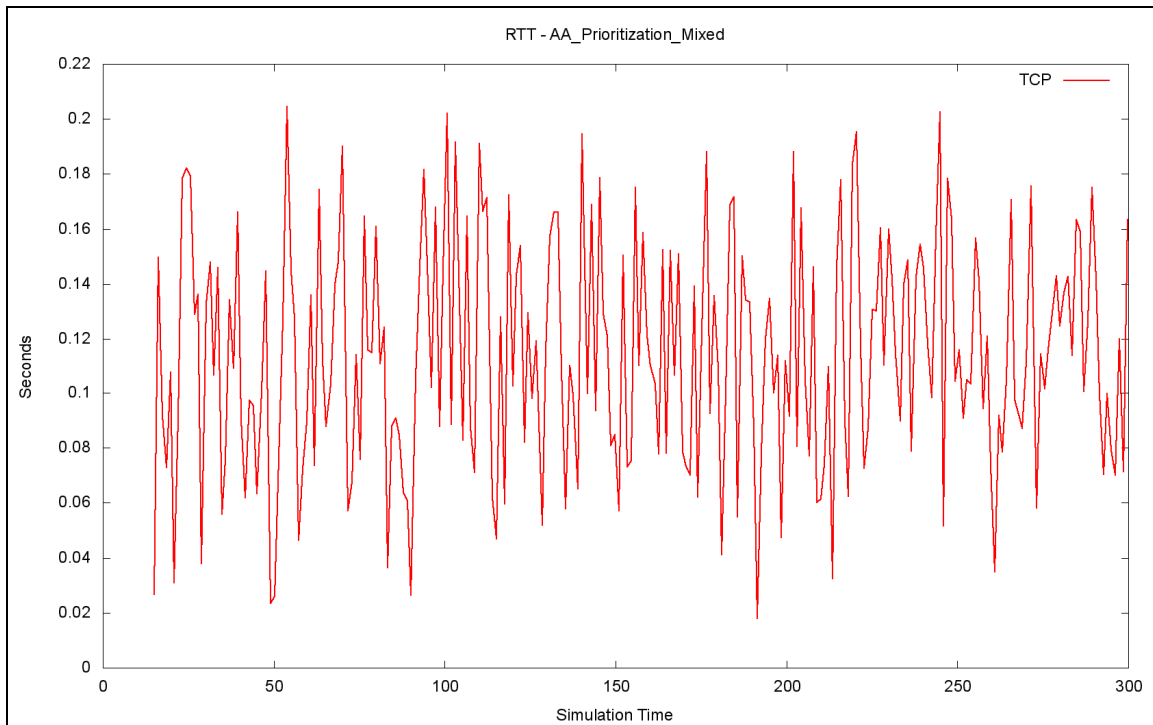


Figure 33 - Round Trip Time, AA Prioritization, Mixed

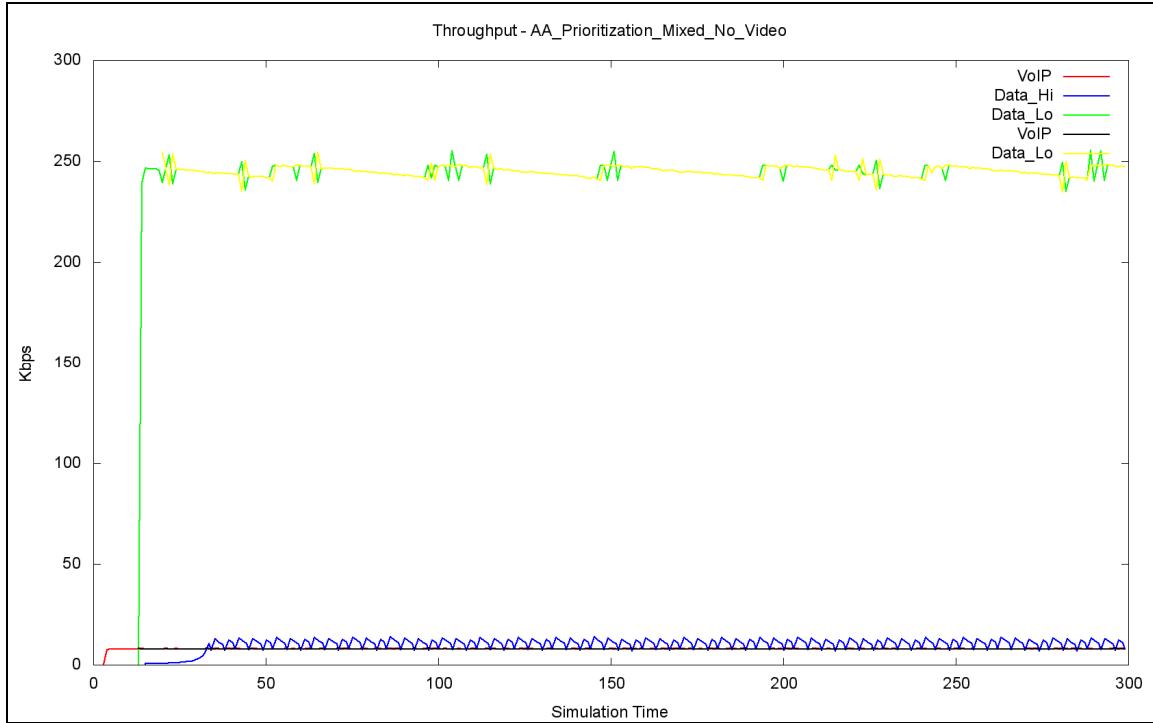


Figure 34 - Throughput, AA Prioritization, Mixed, No Video

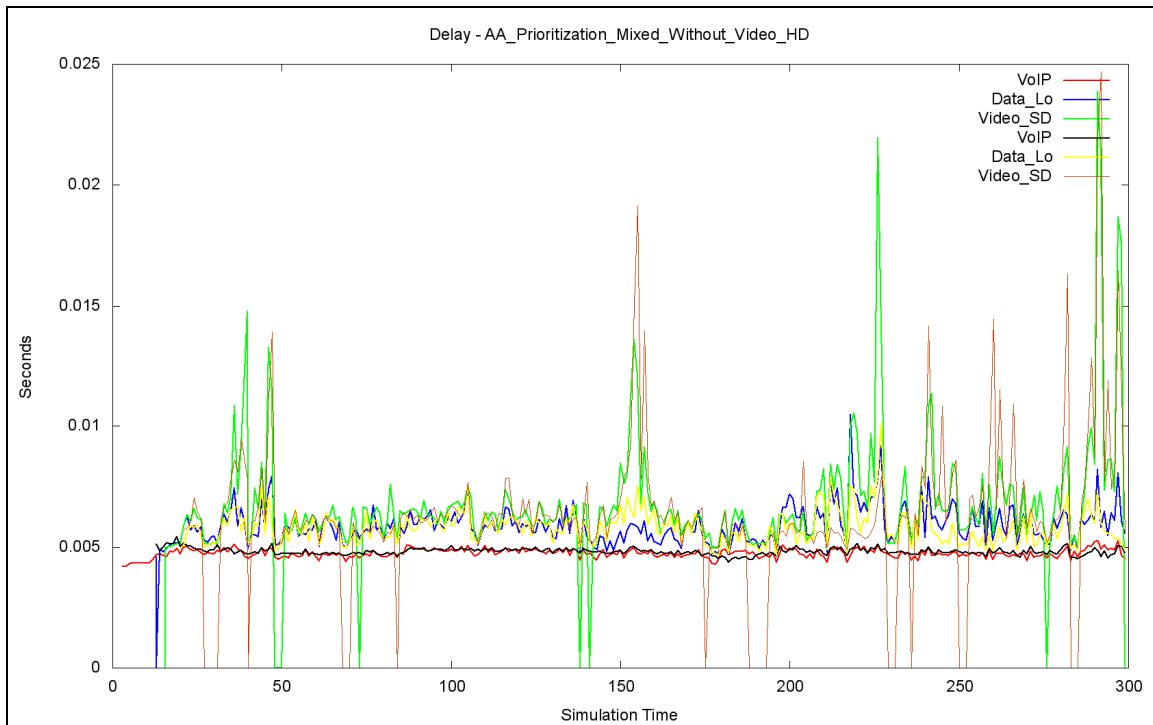


Figure 35 - Delay, AA Prioritization, Mixed, No Video\_HD

## CONCLUSIONS

An extension to 802.11e has been presented that dynamically adapts the contention parameters to achieve prioritization with limited network knowledge and performance similar to static prioritization. The results presented show that dynamic optimization does lead to better overall network utilization in some cases. More importantly, it was shown that queue access delays can be effectively used to get performance feedback in order to make intelligent optimization decisions. The Adaptive Algorithm performs as well, and in some cases better, than static prioritization.

The data presented here shows that the Adaptive Algorithm can be deployed in these scenarios to effectively manage priority without a complicated centralized infrastructure. Simplicity is crucial for rapid adoption in the consumer space. For governments and corporations wishing to adopt 802.11, protection of high priority data from background data is essential.

Although we feel that the algorithm presented here is important, we believe it is not the total solution. It has been shown in previous works that in general 802.11e increases contention in the network, which results in overall lower throughput than without prioritization. The Adaptive Algorithm does alleviate this disadvantage for some scenarios, but not all. Other works show that 802.11e under the HCCA can solve the problem of throughput at the expense of simplicity, and rapid deployment. Therefore, we expect that many 802.11e deployments will use a mix of pure EDCA and HCCA in addition to using EDCA as a backup measure for HCCA. Having an efficient and simple way to use ECDA will provide network architects with greater flexibility and will further reduce dependence on proprietary network designs.

## APPENDIX

A development environment that contains all source code, simulation scripts, and data relevant to the work presented has been created to facilitate further research in this area. The development environment is in the form of a VMware virtual machine that can be used with the free VMware Player. All necessary instructions to begin working with the virtual machine can be found in the included file README.txt. The archive can be found at the following URL: [http://www.clemson.edu/~jmarty/802\\_11e.html](http://www.clemson.edu/~jmarty/802_11e.html). In the event of problems with this URL, please contact either [jim.martin@cs.clemson.edu](mailto:jim.martin@cs.clemson.edu) or [cpssc@cs.clemson.edu](mailto:cpssc@cs.clemson.edu).

## REFERENCES

- [1] ANSI/IEEE Std. 802.11b, 1999 Edition. IEEE Standard for Information technology – Telecommunications and Information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11. IEEE, 1999
- [2] ANSI/IEEE Std. 802.11e, 1999 Edition. IEEE Standard for Information technology – Telecommunications and Information exchange between systems – Local and metropolitan area networks – Specific requirements. Part 11, Amendment 8. IEEE, 1999
- [3] Wi-Fi CERTIFIED for WMM – Support for Multimedia Applications with Quality of Service in Wi-Fi Networks. Wi-Fi Alliance. September 1, 2004.
- [4] Swaminathan, Arvind and James Martin. 2006. “Fairness Issues in Hybrid 802.11b/e Networks”. Consumer Communications and Networking Conference. 2006 3rd IEEE Vol. 1, Issue 8-10. 50-54.
- [5] Wiethölter, Sven and Hoene, Christian. 2003. Design and Verification of an IEEE 802.11e EDCA Simulation Model in ns-2.26. TKN Technical Report Series. Technical University Berlin. November, 2003.
- [6] Cali, F; M. Conti; E. Gregori. 2000, “Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit”. IEEE/ACM Transactions On Networking, Vol. 8, No. 6, December 2000, 785-799.

- [7] Cali, F; M. Conti; E. Gregori. 2000, "IEEE 802.11 Protocol: Design and Performance Evaluation of an Adaptive Backoff Mechanism". IEEE Journal on Selected Areas in Communications. Vol. 18. No. 9 (September).
- [8] Chen, D; S. Garg; M. Kappes; K. Trivedi. 2002. "Supporting VBR VoIP Traffic in IEEE 802.11 WLAN in PCF Mode". Center for Advanced Computing and Communications, ECE Department. Duke University, Durham, NC. Avaya Labs Research, New Jersey.
- [9] Siris, Vasilios and Stamatakis, George. 2006. "Optimal CWmin Selection for Achieving Proportional Fairness in Multi-Rate 802.11e WLANs: Test-bed Implementation and Evaluation". International Conference on Mobile Computing and Networking: WiNTECH 06.
- [10] He, Dajiang and Charles Shen. 2003. "Simulation Study of IEEE 802.11e EDCA". Vehicular Technology Conference, Spring 2003. Vol. 1. 685-689.
- [11] Xiao, Yang; L. Haizhon; S. Choi. 2004. "Protection and Guarantee for Video and Voice Traffic in IEEE 802.11e Wireless LANs". INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communication Societies. Volume 3. Issue 7-11. 2152-2162.
- [12] W. Spearman, J. Martin, "A Distributed Adaptive Algorithm for QoS in 802.11e Wireless Networks", Proceedings of the 2007 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'07), (San Diego, CA, July 2007), pp. 379-386.
- [13] Lim, L.W. Malik, R. Tan, P.Y. Apichaichalermwongse, C. Ando, K. Harada, Y. Panasonic Singapore Labs. "A QoS Scheduler for IEEE 802.11e WLANs". Consumer Communications and Networking Conference, 2004. pp. 199-204.

- [14] V. Vleeschauwer, J. Janssen, G. Petit, and F. Poppe, "Quality bounds for packetized voice transport," Alcatel, Tech. Rep., 1st Quarter 2000.
- [15] ITU – Series H, Audiovisual and Multimedia Systems, Infrastructure of audiovisual services – Coding of moving video, H.264. 03/2005. International Telecommunication Union.
- [16] Garg, S. Kappes, M. Krishnakumar, A. "On the Effect of Contention-Window Sizes in IEEE 802.11b Networks". Avaya Labs Research. June 25, 2002.
- [17] Casetti, C. Chiasserini, C. "Improving Fairness and Throughput for Voice Traffic in 802.11e EDCA". Personal, Indoor and Mobile Radio Communications, 2004. PIMRC 2004. 15th IEEE International Symposium. September, 2004. Volume 1. Pages 525-530.
- [18] Angelis, F. Habib, I. Altieri, P. Giambene, G. "Differentiated QoS Provision for Multimedia Traffic in WiFi Systems". Communications, 2005. ICC 2005. 2005 IEEE International Conference on. Volume 2, May 2005. Pages 1251-1255.
- [19] Leith, D. Clifford, P. "Using the 802.11e EDCF to Achieve TCP Upload Fairness over WLAN Links". Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on. Volume 3. Issue 7. April 2005. Pages 109 – 118.
- [20] Spearman, W. Martin, J. Gao, J. "Adaptive QoS in 802.11e Wireless Networks for Lunar Communication". IEEE Aerospace Conference, Big Sky Montana, March 2008.